



UNIVERSIDAD CARLOS III DE MADRID  
ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE INFORMÁTICA  
DOCTORADO EN CIENCIA Y TECNOLOGÍA INFORMÁTICA

TESIS DOCTORAL

# **Metodología de Simulación Paralela aplicada a Sistemas Multi-Agente en Entornos Virtuales**

Oscar Calvo León

DIRIGIDA POR

José Manuel Molina López

Miguel Angel Patricio Guisado

Mayo de 2017 v4



This work is distributed under the Creative Commons 3.0 license. You are free to copy, distribute and transmit the work under the following conditions: (i) you must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work); (ii) you may not use this work for commercial purposes, and; (iii) you may not alter, transform, or build upon this work. Any of the above conditions can be waived if you get permission from the copyright holder. See <http://creativecommons.org/licenses/by-nc-nd/3.0/> for further details.

---

Web page: <http://raxolabs.duckdns.org/oscar/index.html>

E-mail: [ocalvo@gmail.com](mailto:ocalvo@gmail.com)

Telephone: +34 91 484 99 85

Address:

Grupo de Inteligencia Artificial Aplicada  
Departamento de Informática  
Universidad Carlos III de Madrid  
Av. de la Universidad Carlos III, 22  
Colmenarejo 28270 — Spain

# **Metodología de Simulación Paralela aplicada a Sistemas Multi-Agente en Entornos Virtuales**

**Autor:** Oscar Calvo León

**Directores:** José Manuel Molina López

Miguel Angel Patricio Guisado

Firma del Tribunal Calificador:

Nombre y Apellidos

Firma

Presidente: D. ....

Vocal: D. ....

Secretario: D. ....

Calificación: .....

Colmenarejo, ..... de ..... de 2017.



*a Mónica.*



---

# Índice general

<b>Resumen</b>	<b>xxi</b>
<b>Abstract</b>	<b>xxiii</b>
<b>Agradecimientos</b>	<b>xxv</b>
<b>1 Introducción</b>	<b>1</b>
1.1 Motivación . . . . .	1
1.2 Los problemas de la Realidad Virtual a día de hoy . . . . .	2
1.2.1 Descripción de los Problemas . . . . .	3
1.2.2 Análisis y Categorización de Problemas Encontrados . . . . .	5
1.3 Objetivo . . . . .	7
1.3.1 Definición formal de los Objetivos . . . . .	7
1.3.2 Hipótesis expuestas . . . . .	7
1.4 Propuesta . . . . .	11
1.4.1 G1 - Metáfora antropomórfica adecuada . . . . .	11
1.4.2 G2 - Principios de diseño . . . . .	11
1.4.3 G3 - Formalización abstracta . . . . .	12
1.4.4 G4 - Método formal para medir inmersión . . . . .	12
1.4.5 G5 - Método formal para medir validez ecológica . . . . .	12
1.4.6 G6 - Tipos de elementos y servicios posibles . . . . .	12
1.4.7 G7 - Modelo teórico software específico . . . . .	12
1.4.8 G8 - Metodología específica . . . . .	13

1.4.9	G9 - Definición de servicios necesarios . . . . .	13
1.4.10	G10 - Plataformas de servicios . . . . .	13
1.4.11	G11 - Frameworks de servicios . . . . .	14
1.4.12	Resumen de propuestas . . . . .	14
1.5	Validación . . . . .	15
<b>2</b>	<b>Estado Del Arte</b>	<b>17</b>
2.1	Introducción . . . . .	17
2.2	Estudio sectorial usos e intereses en Realidad Virtual . . . . .	19
2.2.1	Introducción . . . . .	19
2.2.2	La Realidad Virtual en perspectiva . . . . .	19
2.2.3	Surgimiento, caída y resurgimiento . . . . .	25
2.2.4	Breve historia comparativa . . . . .	26
2.2.5	Usos prácticos de la Realidad Virtual . . . . .	27
2.2.6	Sectores e Industrias existentes . . . . .	29
2.2.7	Campos de aplicación de la Realidad Virtual . . . . .	32
2.2.7.1	Automoción . . . . .	32
2.2.7.2	Educación . . . . .	34
2.2.7.3	Entretenimiento . . . . .	36
2.2.7.4	Farmacéuticas . . . . .	36
2.2.7.5	Industria . . . . .	37
2.2.7.6	Juegos y Serious Games . . . . .	38
2.2.7.7	Marketing . . . . .	39
2.2.7.8	Medicina . . . . .	41
2.2.7.9	Militar . . . . .	45
2.2.7.10	Minería . . . . .	48
2.2.7.11	Neurociencia . . . . .	49



2.2.7.12	Neurosicología . . . . .	50
2.2.7.13	Restauración . . . . .	50
2.2.7.14	Retail . . . . .	51
2.2.7.15	Robótica . . . . .	53
2.2.7.16	Transporte . . . . .	54
2.2.7.17	Turismo . . . . .	56
2.2.7.18	Urbanismo . . . . .	57
2.2.8	Sectores de con aplicación . . . . .	59
2.2.9	Casos de Uso identificados . . . . .	59
2.2.9.1	Matriz publicaciones / casos de uso . . . . .	61
2.2.9.2	Las cinco categorías de casos de uso de la Realidad Virtual	64
2.2.9.3	Relaciones y dependencias entre grupos de uso de la Realidad Virtual . . . . .	66
2.2.9.4	Casos de uso VR por Sector Económico . . . . .	68
2.2.9.5	Ranking de sectores por casos de uso de la Realidad Virtual . . . . .	71
2.2.9.6	Evolución sectores y casos de uso de la Realidad Virtual	72
2.2.10	Problemas Identificados . . . . .	74
2.2.10.1	Lista de problemas identificados . . . . .	74
2.2.10.2	Matriz de Publicaciones / Problemas . . . . .	75
2.2.10.3	Análisis de las estadísticas de problemas detectados . .	77
2.2.10.4	Resumen de los problemas no-hardware más importantes	79
2.3	Sistemas Multiagente . . . . .	80
2.3.1	Introducción . . . . .	80
2.3.2	Inicios . . . . .	81
2.3.3	Ferber y Muller . . . . .	81
2.3.4	Bandini, Manzoni y Simone . . . . .	82

2.3.5	Weyns et al. . . . .	85
2.3.5.1	El entorno como contenedor y medio de comunicación .	86
2.3.5.2	El entorno como una capa organizacional . . . . .	86
2.3.5.3	El entorno como infraestructura de coordinación para agentes cognitivos . . . . .	87
2.3.5.4	Entornos de Markov . . . . .	87
2.3.5.5	Entornos de tarea . . . . .	88
2.3.5.6	Responsabilidades del entorno . . . . .	90
2.3.5.7	Niveles de soporte del modelo . . . . .	94
2.3.6	Casos prácticos de MAS Situados . . . . .	96
2.3.6.1	Introducción . . . . .	96
2.3.7	Modelo de Weyns, Schelfthout y Holvoet, 2005 . . . . .	96
2.3.7.1	Funcionalidades . . . . .	96
2.3.7.2	Distribución de las responsabilidades . . . . .	97
2.3.7.3	Descripción del entorno virtual de los agentes AGV . .	98
2.3.7.4	Conclusiones . . . . .	100
2.3.8	Experiencia de Bandini, Manzoni, y Simone, 2001 . . . . .	100
2.3.9	Propuesta de Helleboogh, Vizzarri, Uhrmacher y Michel, 2007 . .	101
2.3.10	Modelo teórico de Wang et al., 2011 . . . . .	103
2.3.11	Planteamiento de Ritchings, Anastassakis y Panayiotopoulos, 2001	104
2.3.12	Experimento de Mamei y Zambonelli, 2004 . . . . .	105
2.3.13	Estudio de Friedman, Steed y Slater, 2007 . . . . .	107
2.3.14	Aproximación de Cranefield y Li, 2010 . . . . .	108
2.3.15	Resumen del estado del arte en Sistemas Multiagente Situados .	109
2.4	Estado del Arte relativo a los objetivos . . . . .	110
2.4.1	Metáfora antropomórfica adecuada . . . . .	110
2.4.2	Principios de diseño . . . . .	111

2.4.3	Formalización abstracta . . . . .	113
2.4.4	Método formal para medir inmersión . . . . .	114
2.4.5	Método formal para medir validez ecológica . . . . .	114
2.4.6	Tipos de elementos y servicios posibles . . . . .	115
2.4.7	Modelo teórico software específico . . . . .	115
2.4.8	Metodología específica . . . . .	115
2.4.9	Definición de servicios necesarios . . . . .	116
2.4.10	Plataformas de servicios . . . . .	116
2.4.11	Frameworks de servicios . . . . .	116
<b>3</b>	<b>Modelo Teórico y Metodología</b>	<b>117</b>
3.1	Introducción . . . . .	117
3.2	Metáfora del Entorno . . . . .	119
3.2.1	Tecnología de soporte . . . . .	119
3.2.2	Principios de Diseño . . . . .	120
3.2.2.1	Validez Ecológica . . . . .	120
3.2.2.2	Principio de Máxima Verosimilitud . . . . .	121
3.2.2.3	Flujo de Compromiso de Presencia . . . . .	122
3.3	Formalización abstracta del modelo . . . . .	124
3.3.1	Descripción formal de entorno . . . . .	124
3.3.1.1	Estados de un Entorno . . . . .	124
3.3.1.2	Objetos de un Entorno . . . . .	125
3.3.1.3	Servicios de un Entorno . . . . .	127
3.3.2	Medidas de similitud entre Servicios . . . . .	128
3.3.2.1	Función de similitud simple . . . . .	129
3.3.2.2	Condiciones de similitud entre conjuntos de servicios . . . . .	129
3.3.2.3	Función de similitud sobreyectiva . . . . .	130

---

3.3.2.4	Función de similitud entre entornos . . . . .	130
3.3.3	Casos particulares de entorno . . . . .	131
3.3.3.1	Entornos euclídeos de grado 3 . . . . .	132
3.3.3.2	Entorno real . . . . .	134
3.3.4	Índice de Inmersión . . . . .	135
3.3.5	Entornos Virtuales Inmersivos . . . . .	136
3.3.6	Índice de Validez Ecológica . . . . .	137
3.4	Taxonomía de Servicios . . . . .	138
3.4.1	Clasificación general de los servicios . . . . .	138
3.4.2	Clasificación de Servicios Ecológicos . . . . .	139
3.4.3	Identificación de servicios ecológicos físicos a partir de la observación del mundo real . . . . .	140
3.4.4	Principios físicos modernos . . . . .	141
3.4.5	La persistencia del entorno . . . . .	142
3.4.6	La impenetrabilidad como principio de la interacción . . . . .	142
3.4.7	Identificación de servicios Perceptivos . . . . .	143
3.4.8	Identificación de servicios Interactivos . . . . .	144
3.4.9	Taxonomía completa de Servicios Ecológicos . . . . .	145
3.4.10	Formalización notación taxonómica . . . . .	145
3.4.11	Tecnología . . . . .	146
3.4.12	Taxonomía de Servicios, conclusiones . . . . .	148
3.5	Modelo arquitectónico DEC2RPM . . . . .	149
3.5.1	El experimento del comportamiento complejo . . . . .	149
3.5.2	Interacciones físicas complejas . . . . .	150
3.5.3	Interacciones inteligentes . . . . .	151
3.5.4	El concepto de Avatar . . . . .	152

3.5.5	Taxonomía de interacciones inteligentes entre elementos según su naturaleza . . . . .	153
3.5.6	Los tres canales de interacción . . . . .	154
3.5.7	Módulo Motor Físico . . . . .	155
3.5.8	Módulo Interprete - Controlador . . . . .	156
3.5.9	Módulo Despliegue - Representación . . . . .	156
3.5.10	Módulo de Soporte de Sistema . . . . .	157
3.5.11	Modelo arquitectónico, conclusiones . . . . .	157
3.6	Modelo Despliegue - Representación . . . . .	159
3.6.1	Objetivos marcados al DRM . . . . .	159
3.6.2	Contexto . . . . .	159
3.6.3	División entre Despliegue y Representación . . . . .	160
3.6.4	Estructura de Despliegue . . . . .	161
3.6.5	Representación Simbólica Espacial . . . . .	162
3.6.6	Esquema de componentes del modelo despliegue - representación	163
3.7	Descripción de la metodología PIIVE-DM . . . . .	164
3.7.1	Preguntas frecuentes sobre esta metodología . . . . .	165
3.8	Pasos de la Metodología . . . . .	165
3.8.1	Paso 1. Definición de Requisitos . . . . .	166
3.8.1.1	Requisitos de Servicio ¿Qué? . . . . .	166
3.8.1.2	Requisitos de Entorno ¿Dónde? . . . . .	168
3.8.1.3	Requisitos de Público ¿Quién? . . . . .	169
3.8.1.4	Requisitos de Despliegue ¿Cómo? . . . . .	170
3.8.2	Paso 2. Arquitectura de Servicios . . . . .	170
3.8.3	Paso 3. Estudio de Despliegue . . . . .	171
3.8.4	Paso 4. Diseño de Tecnología . . . . .	175
3.9	Herramientas de soporte a la metodología . . . . .	177

3.9.1	Hoja de Servicios . . . . .	177
3.9.2	Esquema Taxonomía completa de Servicios . . . . .	178
3.9.3	Diagrama de Módulos DRM . . . . .	179
3.9.4	Correspondencia Taxonomía y DRM . . . . .	180
3.9.5	Diagrama simplificado DRM con Servicios . . . . .	181
3.10	Proveedores de Servicio y Proveedores de Entorno . . . . .	182
3.10.1	Motores de Juegos como proveedores de servicio . . . . .	182
3.10.2	Metaversos como proveedores de entorno . . . . .	183
3.10.3	Roles implicados en la metodología . . . . .	185
3.10.4	Ejemplos de uso de VIEs . . . . .	185
3.11	Metodología PIIVE para Servicios Tecnológicos complejos. PIIVE-STC . . . . .	187
3.11.1	Campo de aplicación . . . . .	187
3.11.2	Implementación de Servicios mediante IA . . . . .	188
3.12	Sistemas Multiagente Situados en Entornos Virtuales Inmersivos . . . . .	190
3.12.1	Estructura de Despliegue . . . . .	192
3.12.1.1	Servicios Internos . . . . .	194
3.12.1.2	Servicios Externos . . . . .	195
3.12.1.3	Servicios Deliberativos Propuestos . . . . .	196
3.12.1.4	Servicios Reactivos Propuestos . . . . .	197
3.12.2	Representación Simbólica Espacial . . . . .	197
3.13	Framework SSR-Hub . . . . .	203
3.13.1	Módulos software . . . . .	203
3.13.2	Características técnicas de la herramienta . . . . .	203
3.13.3	Interfaz gráfico de la herramienta . . . . .	206
3.13.4	Seguridad de la herramienta . . . . .	209
3.14	Conclusiones . . . . .	210

<b>4 Experimentación</b>	<b>211</b>
4.1 Introducción . . . . .	211
4.2 Experimento SL/2009 . . . . .	213
4.2.1 Introducción . . . . .	213
4.2.2 Desarrollos teóricos del experimento SL/2009 . . . . .	214
4.2.2.1 Definición de Metabot . . . . .	214
4.2.2.2 AvatarRank . . . . .	215
4.2.3 Desarrollos prácticos del experimento SL/2009 . . . . .	220
4.2.3.1 Metabot Usher . . . . .	220
4.2.3.2 Metabot Pretender . . . . .	222
4.2.3.3 Metabot Chatterbox . . . . .	223
4.2.4 Conclusiones . . . . .	224
4.3 Experimento SL/2011 . . . . .	227
4.3.1 Introducción . . . . .	227
4.3.2 Estrategia Multiagente Deliberativa . . . . .	228
4.3.3 Arquitectura Sprawl1 . . . . .	229
4.3.3.1 Descripción del problema: Objetivo . . . . .	229
4.3.3.2 Descripción del problema: Actores participantes . . . . .	231
4.3.3.3 Interfaz Web . . . . .	237
4.3.3.4 Recursos y su relación con servicios y actores . . . . .	237
4.3.3.5 Autómata de estados . . . . .	239
4.3.3.6 Descripción de la ontología . . . . .	239
4.3.3.7 Ejemplo de diálogos FIPA-ACL (KQML) . . . . .	242
4.3.4 Experimentación . . . . .	245
4.3.4.1 Problema 1. Pasar por la puerta . . . . .	245
4.3.4.2 Problema 2. El pasillo . . . . .	246
4.3.4.3 Problema 3. El conflicto . . . . .	247

4.3.4.4	Problema 4. La cooperación . . . . .	249
4.3.5	Conclusiones . . . . .	251
4.4	Experimento OS/2012 . . . . .	252
4.4.1	Introducción . . . . .	252
4.4.2	Estrategia Sprawl2: Feromonas, Campos y Tags . . . . .	253
4.4.2.1	Antecedentes . . . . .	255
4.4.3	Descripción del concepto Seed . . . . .	255
4.4.3.1	Función de propagación de la energía . . . . .	257
4.4.3.2	El coeficiente de propagación . . . . .	258
4.4.3.3	La distancia especial . . . . .	259
4.4.3.4	Representación gráfica . . . . .	260
4.4.3.5	Campo de seeds (SeedSets) y medición de valores . . . . .	267
4.4.3.6	Tags y tagargs . . . . .	269
4.4.3.7	Diseño y funcionamiento del Hub . . . . .	271
4.4.4	Experimento 1: Búsqueda y almacenamiento de recursos (ant colony) . . . . .	276
4.4.4.1	Enunciado del problema . . . . .	276
4.4.4.2	Estructura de Despliegue . . . . .	276
4.4.4.3	Resultados . . . . .	280
4.4.5	Experimento 2: Bandadas dinámicas (flocking) . . . . .	281
4.4.5.1	Enunciado del problema . . . . .	281
4.4.5.2	Estructura de Despliegue . . . . .	281
4.4.5.3	Resultados . . . . .	283
4.4.6	Conclusiones . . . . .	283
4.5	Experimento OS/2016 . . . . .	284
4.5.1	Introducción . . . . .	284
4.5.2	Planteamiento iterativo de la experiencia . . . . .	284



4.5.3	Objetivo docente de la experiencia . . . . .	285
4.5.4	Objetivo investigador de la experiencia . . . . .	286
4.5.4.1	Metodología PIIVE-DM . . . . .	286
4.5.4.2	OpenSim como Proveedor de Entorno . . . . .	290
4.5.5	Descripción de la experiencia práctica . . . . .	290
4.5.5.1	Descripción del experimento: . . . . .	291
4.5.5.2	Enriquecer la experiencia: . . . . .	291
4.5.5.3	Análisis de los datos . . . . .	292
4.5.6	Respuesta de los alumnos . . . . .	292
4.5.7	Lecciones aprendidas / toma de decisiones . . . . .	293
4.5.8	Conclusiones . . . . .	294
4.6	Conclusiones . . . . .	296
<b>5</b>	<b>Conclusiones</b>	<b>299</b>
5.1	Recapitulación . . . . .	299
5.2	Principales Aportaciones . . . . .	303
5.3	Futuros Trabajos . . . . .	304
<b>A</b>	<b>Apéndice</b>	<b>307</b>
A.1	Proyectos relacionados . . . . .	307
A.1.1	Babybot . . . . .	307
A.1.2	Laboratorios Virtuales UPM . . . . .	307
A.2	Publicaciones relacionadas . . . . .	308
A.2.1	Publicaciones en revistas internacionales (JCR) . . . . .	309
A.2.2	Publicaciones en conferencias . . . . .	309
A.3	Acceso a la última versión del documento . . . . .	310
	<b>Bibliografía</b>	<b>311</b>



---

## Índice de figuras

1.1	Ciclo de Hype de Gartner y Curva S de Foster . . . . .	2
1.2	Matriz de relación entre problemas y necesidades . . . . .	6
2.1	Evolucion «Virtual Reality» en literatura . . . . .	20
2.2	Evolución en el tiempo de las publicaciones académicas relativas a "Virtual Reality. <sup>en</sup> WebOfKnowledge . . . . .	21
2.3	Evolución «Virtual Reality» en Google Trends . . . . .	22
2.4	Interés estimativo en «Virtual Reality» . . . . .	23
2.5	Interés comparativo «Virtual Reality» . . . . .	24
2.6	Evolución del interés en distintos términos equivalentes . . . . .	26
2.7	Matriz de publicaciones y casos de uso . . . . .	61
2.8	Matriz clusterizada . . . . .	63
2.9	Las cinco categorías de los casos de uso . . . . .	64
2.10	Distribución de casos de uso por categorías . . . . .	65
2.11	Matriz clusterizada por grupo primario y secundario . . . . .	66
2.12	Tabla de relaciones entre grupos . . . . .	67
2.13	Casos de uso por Sector Económico . . . . .	68
2.14	Mapa de Calor casos de uso por sector económico . . . . .	69
2.15	Grupos de Casos de uso por Sector Económico . . . . .	70
2.16	Mapa de Calor Grupo de Casos de uso por Sector Económico . . . . .	70
2.17	Interacciones de las cinco categorías de casos de uso . . . . .	70
2.18	Ranking por sectores de casos de uso RV . . . . .	71
2.19	Casos de Uso en el tiempo . . . . .	72

2.20 Grupos de Casos de Uso en el tiempo . . . . .	73
2.21 Gráfica Grupos de Casos de Uso en el tiempo . . . . .	73
2.22 Matriz expandida de publicaciones y los problemas detectados . . . . .	75
2.23 Matriz reducida de publicaciones y los problemas detectados . . . . .	76
2.24 Matriz de publicaciones y tipos de problema . . . . .	76
2.25 Distribución porcentual por campo de aplicación . . . . .	77
2.26 Distribución porcentual por campo de aplicación excepto hardware . . . . .	77
2.27 Evolución de los problemas en el tiempo . . . . .	78
2.28 Ranking de problemas . . . . .	79
2.29 Estructura SMAS . . . . .	93
2.30 Niveles de soporte del modelo . . . . .	95
2.31 Esquema entorno AGV . . . . .	99
2.32 Tabla de distancia . . . . .	101
2.33 Ejemplo entorno inteligente . . . . .	103
2.34 Esquema cliente servidor . . . . .	105
2.35 Ejemplo CoFields . . . . .	107
2.36 Esquema del modelo Cranefield . . . . .	108
2.37 Metáfora y su tecnología de soporte . . . . .	111
2.38 Apple Macintosh . . . . .	112
3.1 Diagrama de Canales . . . . .	155
3.2 Diagrama de Módulos . . . . .	158
3.3 Diagrama Teórico del modelo DRM de alto nivel . . . . .	160
3.4 Arquitectura en Simulación Paralela . . . . .	161
3.5 Hola de Servicios . . . . .	177
3.6 Diagrama de Módulos DRM . . . . .	179
3.7 Correspondencia explícita entre la Taxonomía de Servicios y el Diagrama de Módulos DRM . . . . .	180

3.8	Diagrama Simplificado DRM con Servicios . . . . .	181
3.9	Arquitectura SSR Multi-HUB con Múltiples Proveedores de Entorno en Simulación Paralela . . . . .	202
3.10	Vista HOME . . . . .	206
3.11	Vista del botón STATS . . . . .	207
3.12	Vista del admin menu . . . . .	207
3.13	Vista de Preferencias y Panel de Control . . . . .	208
3.14	Interfaz de Usuarios . . . . .	208
3.15	Vista del monitoring vser ss . . . . .	209
3.16	PIN / STOP / Not allowed . . . . .	209
4.1	Función AvatarRankScore: representación espacial plana en un espacio cartesiano X-Y . . . . .	218
4.2	Función AvatarRankScore: representación lineal graduada . . . . .	218
4.3	Ejemplos de AvatarRank . . . . .	219
4.4	Metabot Usher . . . . .	221
4.5	Metabot Pretender . . . . .	222
4.6	Metabot Chatterbox . . . . .	223
4.7	Líneas de chat por metabot: usher, chatterbox o pretender . . . . .	224
4.8	Líneas de chat por apariencia humana del avatar . . . . .	225
4.9	Porcentaje de usuarios engañados . . . . .	226
4.10	Ejemplo de una posible configuración física del problema . . . . .	230
4.11	Modelo BDI . . . . .	235
4.12	Ejemplo de interfaz de usuario . . . . .	238
4.13	Autómata de estados de Agente-X . . . . .	239
4.14	AgenteX se registra en el Agente0 . . . . .	243
4.15	AgenteX se mueve sin necesidad de colaboración . . . . .	243
4.16	AgenteX mueve y rompe obstáculo con ayuda de Y y Z . . . . .	244

4.17 AgenteX se mueve pero tiene que replanificar . . . . .	244
4.18 Pasar por la puerta . . . . .	245
4.19 El pasillo . . . . .	246
4.20 El pasillo según la percepción conjunta de los agentes . . . . .	247
4.21 El conflicto, paso 1 . . . . .	247
4.22 El conflicto, paso 2 . . . . .	248
4.23 El conflicto, paso 3 . . . . .	248
4.24 La cooperación . . . . .	249
4.25 La representación de la percepción de los agentes . . . . .	250
4.26 Detalle del momento en el que los agentes colaboran . . . . .	250
4.27 Propagación lineal con propagación cuadrática y energía inicial 10 . . . . .	260
4.28 Sinusoidal con ate. cuadrática, frecuencia 5 y energia inicial 10 . . . . .	260
4.29 Propagación lineal con propagación cuadrática y energía inicial 10 . . . . .	261
4.30 Propagación polinomial sin atenuación y frecuencia 1/10 . . . . .	261
4.31 Propagación singleflank sin atenuación, frecuencia 10 y energía inicial 10 . . . . .	262
4.32 Ejemplos forma de onda lineal . . . . .	263
4.33 Ejemplos forma de onda sinusoidal . . . . .	264
4.34 Ejemplos forma de onda cuadrada . . . . .	265
4.35 Ejemplos forma de onda polinomial . . . . .	265
4.36 Ejemplos forma de onda un solo flanco . . . . .	266
4.37 Matriz de punteros . . . . .	266
4.38 Ejemplo gráfico 2D automatico . . . . .	274
4.39 Representación para Sprawl2 . . . . .	275
4.40 Vista superior del recinto . . . . .	277
4.41 Interior del recinto . . . . .	278
4.42 Autómata de estados de los agentes-hormiga . . . . .	279
4.43 Implementación concreta DRM en SSR-Hub para el proyecto MSO . . . . .	285

---

4.44	VIL: interfaz de usuario donde construir y programar . . . . .	288
4.45	Detalle VIE-MSO: Dos vistas de la «batidora» en el VIL . . . . .	289
4.46	SSR-MSO consola web: interfaz off-VIL para los alumnos . . . . .	292
4.47	Ejemplo de resultado experimental: histograma de eje Z en ángulos de 15°	293
4.48	SSR-ASE Web Console: monitorización de los agentes en tiempo real . .	294
4.49	Tabla de Scoring de Objetivos . . . . .	296
4.50	Scoring Objetivos . . . . .	297
5.1	Las seis propuestas dentro del ciclo teórico - práctico . . . . .	303
5.2	Las seis propuestas con sus componentes más importantes . . . . .	303
A.1	TesisQR . . . . .	310
A.2	Página Web de Oscar Calvo . . . . .	310





---

# Índice de tablas

1.1	Tabla de Objetivos . . . . .	7
2.1	Sectores Importantes por Forbes . . . . .	29
2.2	Sectores Importantes por según OIT . . . . .	29
2.3	Sectores según Portal Europeo PYMES . . . . .	30
2.4	Sectores a estudio . . . . .	31
2.5	Sectores con aplicación VR . . . . .	59
2.6	Agrupación de casos de uso por categorías . . . . .	62
2.7	Jerarquía de casos de uso . . . . .	64



---

## Resumen

**Q**UALQUIER cambio de paradigma es indicador de que un cambio más profundo se está realizando a nivel tecnológico, cultural e incluso social. Cada salto de paradigma tiene una duración distinta, pero todos comparten algo: una vez el cambio ocurre, nada vuelve a ser como antes.

El término “realidad virtual” lleva más de veinte años siendo profusamente utilizado y ya se ha integrado en nuestra cultura tecnológica. Pero su excesivo y a veces frívolo uso ha difuminado su significado, provocando desconfianza.

Habitualmente se han aglutinado distintos conceptos bajo un mismo paraguas: juegos 3D, visión estereoscópica, video 360°, etc. En el camino de adaptación a este paradigma será necesario aclarar conceptos, acotar tecnologías, diseñar metodologías y sobre todo, aprender las nuevas reglas del juego según surjan y adaptarnos a ellas.

El mayor problema encontrado en las primeras experiencias con la realidad virtual ha sido la inmadurez «hardware», pero si lo comparamos con otro cambio de paradigma como la tecnología móvil y observamos la evolución de los smartphones en los últimos diez años puede que esta inmadurez del hardware de realidad virtual no dure mucho más tiempo.

En cualquier caso la realidad virtual no es únicamente un dispositivo físico: el software asociado es tan o incluso más importante. Suponiendo que el hardware RV siga su ciclo de vida habitual: modelos cada vez más asequibles, fabricación a gran escala y saturación del mercado, es más que probable que el hardware adelante en madurez al software.

Parece por tanto lógico dejar a un lado el hardware, ya encaminado su desarrollo, y centrarse en las carencias que el software presenta en el entorno de la realidad virtual. Al estudiar dichas carencias aparecen grandes lagunas: falta de metodologías, de metáforas válidas, de estrategias de diseño, de software base, de arquitecturas y de modelos teóricos que las soporten.

La presente tesis trata de, abstrayéndose de los problemas del hardware, identificar los problemas más importantes, plantear hipótesis para resolver esos problemas, presentar unas

propuestas innovadoras que puedan solucionarlos y mostrar experiencias que refrenden dichas propuestas.

Se propondrán metáforas y principios de diseño que creen espacios contextuales bien identificados para acabar con la terminología imprecisa, se plantearán modelos matemáticos abstractos para soportar taxonomías y arquitecturas específicas, se mostrarán metodologías prácticas para definir buenas prácticas promoviendo su uso en la comunidad software y se aportarán entornos de trabajo basados en inteligencia artificial idóneos para implementar soluciones.

---

# Abstract

ANY paradigm shift is indicative of a deeper change is taking place at the technological, cultural and even social level. Each paradigm jump has a different duration, but all share something: once the change happens, nothing is as it was before.

The term «virtual reality» has been extensively used for over twenty years and has already been integrated into our technological culture. But its excessive and sometimes frivolous use has blurred its meaning, causing distrust.

Usually, different concepts have been merged under one single umbrella: 3D games, stereoscopic vision, 360° video, etc. In the path of adaptation to this paradigm it will be necessary to clarify concepts, to limit technologies, to design methodologies and, most important, to learn and adapt to the new rules in play.

The biggest problems found in the first experiences with virtual reality are about «hardware» immaturity, but if we compare it with another paradigm shift as mobile technology and observe the evolution of smartphones in the last ten years may be that virtual reality immaturity of won't last much longer.

But virtual reality is not just a physical device: related software is as important as hardware, or even more. Assuming VR hardware will follow the usual lifecycle: increasingly affordable models, large-scale manufacturing and market saturation, hardware is more than likely to grow in maturity quicker than software.

So the logical next step is to leave aside the hardware, and focus on the shortcomings that the software presents in the VR environment. When studying this topic, large gaps arises easily: lack of methodologies, valid metaphors, design strategies, basic software, architectures and theoretical models.

The present thesis deals with the problems of software, identifying the most important, proposing hypotheses to solve these problems, presenting innovative proposals that can solve them and show experiences that endorse these proposals.

We will propose metaphors and design principles that create well-defined contextual spaces to eliminate imprecise terminology, abstract mathematical models to support

specific taxonomies and architectures, practical methodologies will be shown to define good practices promoting their use in the software community and will be provided Artificial intelligence-based work environments suitable for practical solution implementation.

---

# Agradecimientos

EL trabajo que se esconde en las páginas que siguen a esta no existiría de no ser por la ayuda de una larga lista de personas. Debo por ello dar gracias a mis directores Miguel Ángel y José Manuel por su paciencia, por la calidad extrema de sus observaciones y por sus ánimos incondicionales; han sido muchos años de constancia y dedicación.

También debo agradecer a mi familia sus ánimos y apoyo. A mi mujer, gracias por tu infinita paciencia, por tu compañía en nuestro «laboratorio» durante las largas horas de investigación, por prestarme tu sabiduría plástica y tu gusto estético, cosas de las que yo carezco completamente. A mis padres por haber confiado siempre en mí y no haberme puesto más metas que las que yo mismo quisiera ponerme.

Gracias a mis colegas del UPETC ThinkTank, muchas de las ideas que están plasmadas en esta Tesis surgieron de nuestras charlas, ¡y todavía quedan muchas otras!

Gracias también a mis compañeros del curso de Máster en Ciencia y Tecnología Informática 2011-2012 que fueron los primeros conejillos de indias de los laboratorios virtuales de simulación.

Al equipo de los 3DLabs del GATE, auténticos profesionales y virtuosos de la virtualidad y por supuesto a Luis Catalán por contar siempre conmigo.

Por último y de manera especial a Ángel, quiero darte las gracias por ser mi mentor, mi consejero y mi valedor, además de una de las personas más altruistas y generosas que conozco. Sin tu apoyo no habría empezado este camino. Muchas gracias y mucho ánimo.

*Oscar Calvo León  
Madrid, Mayo de 2017.*





---

# 1

## Introducción

### 1.1 Motivación

El título de esta tesis es **Metodología de Simulación Paralela aplicada a Sistemas Multi-Agente en Entornos Virtuales**. Según la Real Academia de la Lengua Española:

**Metodología:**

1. *f. Ciencia del método.*
2. *f. Conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal.*

La motivación de todo este trabajo es, precisamente, la búsqueda de una metodología que permita dar madurez a un proceso puramente artesanal. Este proceso no es otro que la construcción de sistemas computacionales que, siendo capaces de aprovechar los avances en el hardware comúnmente denominado de «Realidad Virtual», permitan la implementación de **Entornos Virtuales Inmersivos**. A lo largo de toda esta tesis se analizarán los problemas de esta tecnología desde el punto de vista del *software*, buscaremos las causas y las posibles soluciones. En base a estas premisas se realizarán una serie de propuestas tanto teóricas como prácticas que serán puestas a prueba posteriormente.

El resultado final es un compendio de paradigmas, principios, metáforas, modelos y buenas prácticas que hemos recogido bajo el término «metodología», y con el objetivo de resolver un amplio conjunto de problemas y carencias que la **Realidad Virtual**, como cualquier otra tecnología emergente e inmadura, padece.

## 1.2 Los problemas de la Realidad Virtual a día de hoy

Tal y como se definirá en el punto 2.2.2 parece evidente que la tecnología comúnmente denominada **Realidad Virtual** está comportándose según el modelo *Gartner Hype Cycle* iterativo (Fenn and LeHong, 2011; Linden and Fenn, 2003). Hasta ahora todo el interés que tanto los medios de comunicación como las industrias han puesto en esta tecnología se ha centrado en el desarrollo de dispositivos que permitan aumentar la experiencia de usuario, lo cual es lógico.

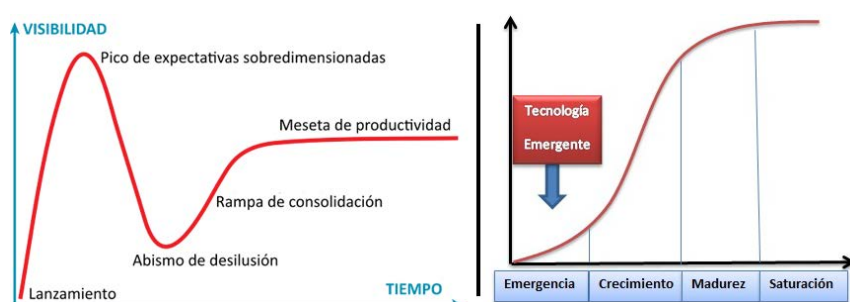


Figura 1.1: Izquierda: Ciclo de Hype de Gartner, representación gráfica de la madurez, adopción y aplicación comercial de una tecnología específica. Derecha: Curva S de Foster, Clasificación de las tecnologías según su madurez

Según el modelo de madurez tecnológica *curva S de Foster* (Foster, 1986) existen cuatro fases: *Emergente*, *Crecimiento*, *Madurez* y *Saturación*. Siguiendo su definición los dispositivos, tanto de realidad virtual como de realidad aumentada, están entrando en la fase de *Crecimiento*, y aunque el modelo de Foster define que la longitud de cada fase es distinta para cada tecnología, algunos autores postulan que en menos de diez años entraremos en fase de *Madurez*.

Tanto el ciclo «*Hype*» de Gartner como la *Curva S de Foster* son aplicables a cualquier tecnología y al proceso industrial productivo asociado, independientemente de que el objeto a producción sea físico (*hardware*) o lógico (*software*). Sin embargo históricamente se ha prestado más atención a las tecnologías «tangibles», por ejemplo en la última edición de la curva de Gartner <http://www.gartner.com/newsroom/id/3412017> se muestran 19 tecnologías *hardware* frente a las 12 tecnologías *software*.

La naturaleza intangible de las aplicaciones y su ciclo de vida mucho más rápido pueden hacer presuponer que una vez el hardware existe, el software aparece automáticamente. Esta suposición, si bien ha sido históricamente cierta durante años, solamente es aceptable cuando la complejidad del sistema es baja y desde una perspectiva de *usuario*. Sin embargo

cuando intentamos generar soluciones de industria nos encontramos reiteradamente con la *Crisis del Software* tal y como la enunció Friedrich L. Bauer en 1968 y Dijkstra formalizó (Dijkstra, 1972).

Teniendo en cuenta estas aclaraciones preliminares podemos definir el **problema** que nos ocupa como ***la no existencia un modelo teórico y metodológico para la construcción de soluciones software V.R que cumpla con unos requerimientos mínimos de portabilidad, escalabilidad, reusabilidad y madurez***, aunque en los siguientes capítulos y a lo largo de la presente tesis estudiaremos, matizaremos y factorizaremos la naturaleza de dicho problema.

### 1.2.1 Descripción de los Problemas

Como se podrá observar con mayor detenimiento más adelante, en el punto 2.2.10.4 se identificarán los problemas más importantes para la **V.R** plasmándolos en este esquema:

Software	$\left\{ \begin{array}{l} \textit{interacción inmadura} \\ \textit{percepción inmadura} \\ \textit{falta de metáforas antropomórficas} \end{array} \right.$
Metodología	$\left\{ \begin{array}{l} \textit{proceso de ingeniería inmaduro} \\ \textit{falta metodología} \\ \textit{validez ecológica cuestionable} \end{array} \right.$
Servicios	$\left\{ \begin{array}{l} \textit{falta de compatibilidad de la plataformas} \\ \textit{falta facilidad acceso para público no experto} \\ \textit{falta de una plataforma genérica, accesible y unificada} \end{array} \right.$

Desde un punto de vista más coloquial, a través de los numerosos estudios y publicaciones examinados, se recogen las siguientes observaciones:

- Los problemas asociados al **hardware** son los más visibles y los primeros que son detectados, pero se escapan al alcance de la presente tesis y vamos a centrarnos en el resto. En cualquier caso, en el punto 2.2.10.3 se detallan brevemente cuáles son dichos problemas.

- La mayor parte de problemas restantes son de dos tipos: **software** o **metodológicos**
- El **software** se percibe como inmaduro, particularmente en lo relativo a los procesos interactivos y perceptivos. Las experimentaciones (Berg and Vance, 2016; Stork et al., 2012) quedan incompletas y los usuarios no obtienen una buena capacidad inmersiva.
- En lo tocante a la **metodología**, existe una falta de consenso en la terminología y sus significados, por ejemplo los términos «inmersivo, entorno, virtual, etc.» que se utilizan ampliamente (Tussyadiah et al., 2016) no están definidos formalmente y se utilizan de manera intuitiva
- También se detecta (Rothbaum et al., 2014; Tussyadiah et al., 2016) una necesidad de disponer de metáforas antropomórficas y principios de diseño que ayuden a mejorar la inmersión final
- La falta de una metodología es sin duda un gran obstáculo para la formalización de cualquier disciplina (Kim et al., 2005; Zyda, 2005)
- La «validez ecológica» se utiliza profusamente para indicar la credibilidad y efectividad de las técnicas **V.R** a lo largo de prácticamente todos los sectores (medicina, turismo, industria,..) y parece que se está implantando como un estándar «de facto» sin embargo no existe un método que permita evaluar este índice, siendo en todos los casos un valor estimativo
- En lo relativo a los **servicios**, si bien es cierto que los problemas son menos acuciantes, se han detectado una falta generalizada de los mismos
- En particular se denuncia reiteradamente la dificultad para los «no expertos en **V.R**» embarcarse en estos proyectos, tanto a la hora de construirlos como de extraer información de los mismos.

### 1.2.2 Análisis y Categorización de Problemas Encontrados

Teniendo toda esta información vamos a identificar las causas en forma de carencias concretas que provocan dichos problemas. O expresado de otra forma, vamos a realizar el camino inverso e identificar qué elementos serían necesarios para mitigar esos problemas. Son los siguientes:

- No existe una metáfora antropomórfica adecuada
- No existen principios de diseño que puedan ser seguidos
- No existe una formalización abstracta o matemática
- No existe una manera formal de medir las características de los entornos virtuales como: inmersión, validez ecológica, etc.
- No existe una norma sobre los tipos de elementos y servicios que pueden ofrecerse dentro de los entornos virtuales
- No existe un modelo software específico en el que basarse para disponer de un proceso industrial propio de una ingeniería madura
- No existe una metodología específica propia de provea de los procedimientos y herramientas necesarias
- No se han definido los servicios necesarios
- No se han propuesto plataformas de servicios
- No se han propuesto frameworks de servicios

Los «problemas» y las «necesidades» se relacionan según esta matriz de trazabilidad:

Problema / Necesidad	metáfora antropomórfica adecuada	principios de diseño	formalización abstracta	método formal para medir inmersión	método formal para medir validez ecológica	tipos de elementos y servicios posibles	modelo teórico software específico	metodología específica	definición de servicios necesarios	plataformas de servicios	frameworks de servicios
interacción inmadura	X	X	X			X					
percepción inmadura	X	X	X	X		X					
falta de metáforas antropomórficas	X	X		X	X	X	X				
proceso de ingeniería inmaduro			X		X		X	X	X		
falta metodología		X	X		X		X	X	X		
validez ecológica cuestionable			X	X	X		X				
falta de compatibilidad de la plataformas		X		X	X	X	X	X	X	X	X
falta facilidad acceso para público no experto								X	X	X	X
falta de una plataforma genérica, accesible y unificada									X	X	X

Figura 1.2: Matriz de relación entre problemas y necesidades

## 1.3 Objetivo

El objetivo de la presente tesis es, partiendo de un conjunto de hipótesis dado, verificar si dichas hipótesis son válidas o no. Estas hipótesis se formularán en función de cada objetivo concreto, condicionando el cumplimiento de la misma a la consecución del objetivo.

Definamos en este momento los objetivos partiendo de la anterior matriz 1.2 como siguen:

### 1.3.1 Definición formal de los Objetivos

Habiendo quedado por tanto establecida la relación causa-efecto entre los «problemas» y las «necesidades» vamos a tomar dicho conjunto de elementos necesarios y convertirlos en los objetivos a conseguir, quedando reflejados en la siguiente tabla como «goals»:

GOAL	Descripción
G1	metáfora antropomórfica adecuada
G2	principios de diseño
G3	formalización abstracta
G4	método formal para medir inmersión
G5	método formal para medir validez ecológica
G6	tipos de elementos y servicios posibles
G7	modelo teórico software específico
G8	metodología específica
G9	definición de servicios necesarios
G10	plataforma de servicios
G11	framework de servicios

Tabla 1.1: Tabla de Objetivos

### 1.3.2 Hipótesis expuestas

Se proponen las siguientes hipótesis iterativas y acumulativas relacionadas con los objetivos:

**Hipótesis 1 (metáfora antropomórfica adecuada)** *Si se dispusiera de una metáfora de diseño adecuada a los entornos virtuales (V.I.E) y en general aplicable a la V.R*

*sería posible mejorar la interactividad de estos entornos y consecuentemente mejorar la experiencia general.*

**Hipótesis 2 (principios de diseño)** *Si existieran unos principios de diseño que, dando soporte a la metáfora antropomórfica de 1, caracterizasen buenas prácticas y usos correctos del potencial de la tecnología se abundaría en una mejora sustancial de la capacidad inmersiva y en general de la experimentalidad y la validez ecológica.*

**Hipótesis 3 (formalización abstracta del entorno)** *Partiendo de la base intuitiva de que cualquier formalización abstracta de un sistema redundaría en ventajas como reusabilidad, independencia, desacoplamiento y encapsulación, planteamos la hipótesis de si existiera una base teórica abstracta del concepto de «entorno» podríamos abrir la puerta a la exploración en un alto nivel de abstracción de los límites del modelo.*

**Hipótesis 4 (método formal para medir inmersión)** *Apoyándonos en la existencia de una formulación abstracta de la hipótesis 3 planteamos que un mecanismo formal y también abstracto capaz de medir el potencial inmersivo de un entorno sería clave a la hora de buscar los elementos que más nos interesen en pro de aumentar este nivel, así como de caracterizar cuantitativamente la calidad inmersiva de cualquier entorno.*

**Hipótesis 5 (método formal para medir validez ecológica)** *Apoyándonos también en la existencia de una formulación abstracta de la hipótesis 3 y de manera similar a la hipótesis 4 planteamos que de disponer de una mecanismo generalizado, abstracto y aplicable a cualquier entorno que nos permitiese evaluar la validez ecológica nos permitiría eliminar cualquier asomo de duda sobre dicha validez, así como optimizar los **V.I.E** buscando maximizar este indicador.*

**Hipótesis 6 (tipos de elementos y servicios posibles)** *Si, además de los mecanismos teóricos propuestos en las hipótesis 3, 4 y 5, pudiésemos contar con una suerte de taxonomía de elementos y servicios aplicables a cualquier **V.I.E** y que siendo demostrable su necesidad y utilidad, tendríamos asegurados los cimientos de cualquier implementación tecnológica real siendo fiel al máximo al modelo teórico subyacente, asegurando que dichas implementaciones cumplirán los criterios de madurez propios de los procesos de ingeniería arraigados en una base teórica consistente.*



**Hipótesis 7 (modelo teórico software específico)** *Sobre la hipótesis 6 y teniendo en cuenta la naturaleza de los servicios identificados, podemos definir la presente hipótesis según la cual si dispusiéramos de un modelo teórico directamente aplicable al análisis de requerimientos, procesos, usos y datos lógicos y relativos a la Ingeniería del Software aplicable a la implementación de cualquier V.I.E estaríamos asegurando no sólo la madurez del sistema, sino el proceso industrial del mismo y la consolidación de la tecnología.*

**Hipótesis 8 (metodología específica)** *Una vez más, partiendo como punto inicial de la hipótesis 7, si dispusiéramos de una metodología específicamente diseñada para la construcción de V.I.E's, en base tanto a un modelo abstracto como a la experiencia en la implementación de dichos sistemas, estaríamos permitiendo reducir el tiempo de construcción, reducir también los fallos de ejecución, mejorar la experiencia final de usuario y aún más importante, universalizando el conocimiento de la tecnología haciendo más sencilla la construcción de sistemas V.I.E sin la necesidad de disponer grandes cantidades de experiencia previa especializada.*

**Hipótesis 9 (definición de servicios necesarios)** *Si, como extensión de la metodología indicada en la hipótesis 8, contásemos con una taxonomía de los servicios que es habitual implementar, así como una guía de buenas prácticas y casos comunes, se mejoraría y facilitaría la labor de ingeniería relativas a la construcción de sistemas V.I.E.*

**Hipótesis 10 (plataformas de servicios)** *Si existiesen plataformas de servicios especialmente adaptadas a las necesidades de los V.I.E que permitiesen la implementación de sistemas reutilizando la mayor parte del sistema, los diseñadores podrían centrarse únicamente en el valor añadido en cada caso, optimizando esfuerzos y minimizando costes, lo cual sin duda redundaría en una mejora general de los sistemas y una mejora en la experiencia de usuario. Paralelamente y enfocándonos a un escenario de madurez del servicio, la existencia de estas plataformas es fundamental para que el público general pueda disfrutar de esta tecnología de manera sencilla.*

**Hipótesis 11 (framework de servicios)** *Por último, y centrándonos en la implementación de servicios en una hipotética situación descrita en 10, la existencia de frameworks software sería el último paso en el camino hacia una madurez completa de la tecnología, el modelo y la metodología. La existencia de estos frameworks permitiría reducir al mínimo los costes de implementación y maximizar la potencia de estos sistemas, asegurando*

*la accesibilidad universal a los servicios a cualquiera, tanto como consumidor o como generador de contenidos y servicios.*

## 1.4 Propuesta

La propuesta de esta tesis consistirá, tras exponer el lógico estudio del Estado del Arte en las investigaciones y la tecnología actuales, en exponer un conjunto de paradigmas, principios, metáforas, modelos y buenas prácticas. Todos estos elementos estarán orientados a solventar los problemas detectados y a cubrir un objetivo concreto.

Una vez expuesto el problema, subdividido en distintas facetas, identificada su relación con la carencia de elementos necesarios y propuesto una serie de objetivos que suplan dicha carencia, estamos en condiciones de describir detalladamente nuestra propuesta.

Por lo tanto, **yo propongo** construir una serie de elementos de tipo teórico, abstracto, metodológico y práctico que cubran las necesidades expuestas. Para ilustrar dicha propuesta de una manera ordenada vamos a enumerar uno a uno todos los objetivos identificados en el punto 1.3.1 explicando cuál es mi propuesta de manera resumida para posteriormente, y a lo largo de todo el capítulo 3 definir completamente el detalle de cada propuesta.

### 1.4.1 G1 - Metáfora antropomórfica adecuada

La metáfora que propongo se denominará **«metáfora del entorno»** y se describe en profundidad dentro del punto 3.2.

### 1.4.2 G2 - Principios de diseño

Se proponen los siguientes principios de diseño:

1. Validez Ecológica
2. Principio de Máxima Verosimilitud
3. Flujo de Compromiso de Presencia

Estos principios se basan en la **metáfora del entorno** y siguen la filosofía *a menor esfuerzo que cualquier persona debe utilizar para comprender un concepto, mayor éxito a la hora de utilizar dicho concepto*.

### 1.4.3 G3 - Formalización abstracta

Mi propuesta es definir un concepto clave como es el «**entorno**» de manera abstracta y matemática tal y como se puede observar en el punto 3.3.1.

### 1.4.4 G4 - Método formal para medir inmersión

El concepto de «**inmersión**» es fundamental para comprender todo el modelo. Mi propuesta es disponer de varias herramientas de base matemática que permitan caracterizar este concepto tal y como se ha definido en 3.3.4.

### 1.4.5 G5 - Método formal para medir validez ecológica

Al igual que el caso del concepto «**inmersión**», se ha formulado «**validez ecológica**» de tal manera que se dispone de herramientas matemáticas para medir este concepto tal y como se muestra en 3.3.6.

### 1.4.6 G6 - Tipos de elementos y servicios posibles

Mi propuesta en este sentido se basa en definir una base de conocimiento en forma de taxonomía de servicios. No solo se propondrá una taxonomía jerárquica, si no que se argumentará su existencia en base a la «metáfora del entorno» tal y como se describe en 3.4.

### 1.4.7 G7 - Modelo teórico software específico

Mi propuesta como modelo arquitectónico teórico sobre el cual poder construir sistemas **V.R** tiene dos vertientes:

1. Un modelo de ámbito total y más general que se denominará «**DEC2RPM**» y está definido en 3.5.
2. Un modelo de ámbito más local, que es un subconjunto del **DEC2RPM** y que se denominará «**DRM**», estando definido a su vez en 3.6.

### 1.4.8 G8 - Metodología específica

Ya se ha argumentado suficientemente la necesidad de una metodología relativa al diseño e implementación de sistemas **V.R.** En este sentido mi propuesta es la «**Metodología PIIVE-DM**» tal y como queda descrita en el punto 3.8.

### 1.4.9 G9 - Definición de servicios necesarios

Basándonos en las arquitecturas **DEC2RPM** y **DRM** disponemos de un modelo completo para la implementación de sistemas **V.R.**, pero además de unos modelos teóricos es necesario definir una estrategia en lo relativo a qué paradigma de programación puede ser el idóneo en este tipo de proyectos.

Es posible resumir mi propuesta en una sola frase que recoge el ánimo y las premisas que la contienen: «los Sistemas Multi-Agente son el paradigma software óptimo para construir servicios en **V.I.E's**. De entre los distintos tipos de **MAS** que existen, los denominados **S.M.A.S** son los que mejor se adaptan a las necesidades del entorno».

Queda por tanto marcado el camino de esta estrategia que será desarrollado al completo en el punto 3.12.

### 1.4.10 G10 - Plataformas de servicios

Mi propuesta en este sentido tiene dos vertientes:

1. Por un lado, no es necesario implementar todo el modelo una y otra vez para cada nuevo proyecto, más al contrario. Mi propuesta en este sentido es establecer una diferenciación clara entre proveedores de entorno y proveedores de servicio, además de otras consideraciones sobre el software actual, quedando todo ello recogido en el punto 3.10.
2. Por otro lado, propongo definir una sub-metodología práctica orientada a lo que he denominado «servicios complejos» o **PIIVE-STC** para abreviar, tal y como se describe en el punto 3.11. Para que exista un ecosistema válido de plataformas de servicios, dichas plataformas deben ofrecer una variedad de servicios uniformes y estándares.

Ambas propuestas son coherentes y en realidad persiguen el mismo objetivo: la estandarización del software.

### 1.4.11 G11 - Frameworks de servicios

Mi propuesta para cubrir esta necesidad se sustenta en dos soluciones software desarrolladas por mí en lenguaje Java:

1. La librería «**MetabotLib**» sobre la cual se han definido gran parte de los experimentos con los que se valida el modelo.
2. La aplicación «**SSR-Hub**» que siguiendo el modelo **DRM** definido en el punto 3.6 se describe en el punto 3.13.

### 1.4.12 Resumen de propuestas

En la siguiente lista se resumen de manera esquemática las propuestas basadas en las hipótesis una vez han sido caracterizadas con mi propuesta específica. De esta manera se podrá referenciar de manera más sencilla cada propuesta según su codificación GNx.

- **G1** - met. ant.: Metáfora del Entorno
- **G2a** - pr. dis.: Principio de Máxima Verosimilitud
- **G2b** - pr. dis.: Validez Ecológica
- **G2c** - pr. dis.: Flujo de Compromiso de Presencia
- **G3** - for. abs.: Entorno  $E=(S,O,F)$
- **G4** - mfm: Índice de Inmersión
- **G5** - mfm: Índice de Validez Ecológica
- **G6** - tip: Taxonomía de Servicios
- **G7a** - mts: Modelo **DEC2RPM**
- **G7a** - mts: Modelo **DRM**
- **G8** - metodología **PIIVE-DM**
- **G9** - def. serv.: Paradigma **S.M.A.S**
- **G10** - plat. serv: Met Práctica **PIIVE-STC**
- **G11a** - framework: **MetabotLib**
- **G11b** - framework: **SSR-Hub**

## 1.5 Validación

Posteriormente se expondrán una serie de experiencias reales en el campo de la investigación, la educación y la inteligencia artificial en los que se han utilizado las propuestas anteriormente citadas. Se proponen **cuatro** experimentos:

1. En **SL/2009** se definirá el concepto de **Metabot** y se presentará el algoritmo **AvatarRank** capaz de medir la cantidad de atención relativa que está teniendo un conjunto de avatares concretos. Basándose en este algoritmo se experimentará con tres Metabots (Usher, Pretender y Chatterbox) con distintas capacidades interactivas para observar qué factores son clave a la hora de medir la inmersión y el compromiso de presencia.
2. En **SL/2011** se propondrá una estrategia multiagente deliberativa orientada a la resolución de problemas y comportamientos complejos. Se definirá un marco de resolución de problemas y un conjunto de experimentos de distinta complejidad. Mediante esta experimentación se podrá observar que la arquitectura es operativa.
3. En **OS/2012** se propone una estrategia multiagente mixta que se denominó «sprawl2» orientada a la resolución de cualquier tipo de problemas. Para construir un marco genérico de resolución de problemas se definió la herramienta software «seed» con la que se resolverán dos problemas clásicos: «ant colony» y «bird flocking». Finalmente se describirán los resultados obtenidos, quedando probada la potencia de la arquitectura y del modelo propuesto.
4. En **OS/2016** nos centraremos en documentar las metodologías propuestas para la construcción de entornos virtuales inmersivos «**PIIVE-DM**» y «**PIIVE-STC**». Para ello se utilizará dichas metodologías en la elaboración del laboratorio virtual de una asignatura de *Máster* que se está realizando actualmente en la *Universidad Carlos III de Madrid*. Se describirá el doble objetivo docente y de investigación de la experiencia. Se documentará el resultado de aplicar la metodología y se describirán los detalles de la experiencia práctica y del experimento docente. Por último se recapitularán las lecciones aprendidas y se evaluará la idoneidad de la metodología para la experimentación propuesta.

Todos estos experimentos, así como la validación de las hipótesis enunciadas están descritos en el capítulo 4.





---

# 2

## Estado Del Arte

**Tercera Ley de Clarke:** *Cualquier tecnología lo suficientemente avanzada es indistinguible de la magia.*

Arthur C. Clarke

### 2.1 Introducción

El presente capítulo tiene el objetivo de recopilar toda la investigación previa que ha servido como base para el modelo teórico y la metodología que se propone. Los libros, artículos y publicaciones que se han estudiado pueden agruparse en dos campos de investigación que, aunque parezcan muy distintos, están relacionados, como se demostrará a lo largo de esta tesis.

Por un lado vamos a estudiar e intentar entender un fenómeno tecnológico, económico y social que a falta de un mejor nombre por ahora denominaremos «Realidad Virtual». Repasaremos sus fundamentos en busca de los campos de utilidad e interés que tiene esta tecnología en función de sus usos y aplicaciones en distintos sectores de actividad económica. El objetivo de esta reflexión es contextualizar el interés académico e industrial en la materia y también discernir si tiene un interés real o es únicamente una moda pasajera.

Desde una perspectiva más técnica y académica se profundizará en el estudio de una materia bien conocida en el ámbito de la Inteligencia Artificial: los Sistemas Multi Agente, y en concreto los Sistemas Multiagente Situados, que son un tipo específico de

agente con unas características muy interesantes y aplicables a los Entornos Virtuales. Es precisamente esta idoneidad que los Sistemas Multiagente Situados presentan para resolver problemas en Entornos Virtuales lo que hace fundamental este repaso previo.

Después, y tomando como base las publicaciones y estudios presentados en el primer punto de este capítulo y demostrada previamente la relevancia de esta tecnología, estudiaremos cuáles son los **problemas** a los cuales se enfrenta la «Realidad Virtual» y que son hoy en día un **hándicap** para su desarrollo real.

Por último vamos a establecer cuáles son los enfoques y estrategias que se han desarrollado para resolver este problema y otros similares hasta la fecha.

## 2.2 Estudio sectorial usos e intereses en Realidad Virtual

### 2.2.1 Introducción

A lo largo del presente estudio analizamos la tecnología que comúnmente se denomina «Realidad Virtual». Primero investigaremos el interés general en la materia. Posteriormente revisaremos comparativamente su historia desde hace 25 años hasta hoy en día. También investigaremos sobre los sectores e industrias actuales para entender qué aplicaciones prácticas han existido en cada uno de ellos. Por último se identificarán los casos de uso transversales y se presentará un ranking de su potencial.

### 2.2.2 La Realidad Virtual en perspectiva

Durante los últimos años se ha detectado un creciente interés general en la sociedad sobre la tecnología que comúnmente se ha dado en llamar «Realidad Virtual». Es difícil dar una definición exacta y precisa de este concepto y es incluso más difícil aún encontrar dos autores que la definan de igual manera. Siendo lo más concisos, inclusivos y generalistas posibles podríamos aventurar la definición de realidad virtual como: *el conjunto de tecnologías mediante las cuales los seres humanos pueden experimentar sensorialmente un entorno distinto del directamente perceptible*

Toda esta tecnología relativa a la «Realidad Virtual» no es precisamente reciente y prueba de ello es esta búsqueda en Google Books Ngram Viewer:

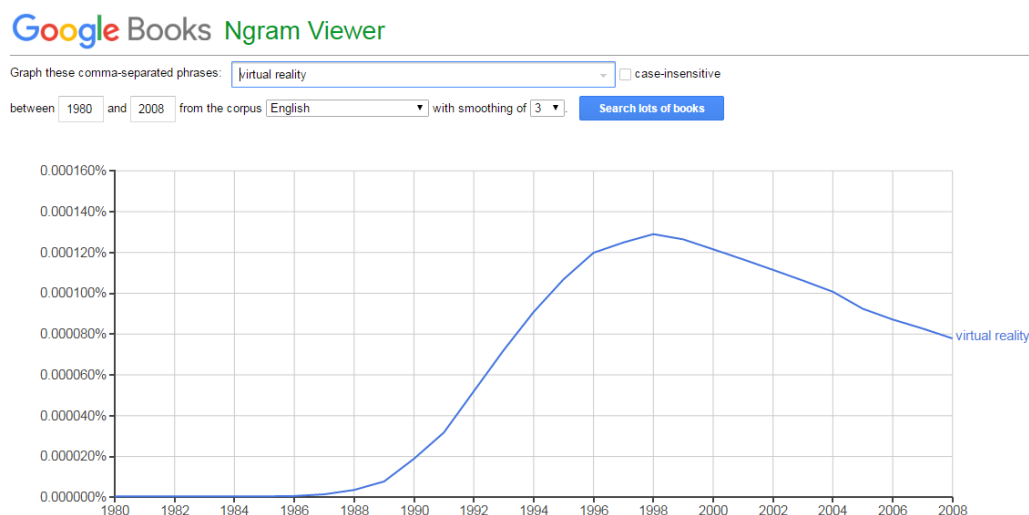


Figura 2.1: Evolucion «Virtual Reality» en literatura

<https://books.google.com/ngrams/graph?content=virtual+reality>

Como puede observarse, el concepto «realidad virtual» comienza a aparecer en la literatura inglesa a finales de los años 80 del siglo pasado. Sin duda, ciertas obras de ciencia ficción clave como «*Neuromante*» (Gibson, 1984) o «*Snow Crash*» (Stephenson, 1992) marcaron una década de un nuevo subgénero literario dentro de la ficción.

Aparte de la pura ficción, la realidad virtual y todos los avances tecnológicos relacionados con esta disciplina han sido objeto de estudio en la comunidad científica desde los años 90. Prueba de ello son los numerosos artículos publicados (según Google Scholar, sólo en la década de los 80 se publicaron unos 1.390 artículos relativos al tema «virtual reality») o la publicación de Springer «Virtual Reality» que comenzó su andadura en 1995 <http://link.springer.com.strauss.uc3m.es:8080/journal/10055>

Sin embargo la prueba más palpable del absoluto y creciente interés académico en esta materia se puede obtener fácilmente a través del <https://www.webofknowledge.com/>. Al lanzar la búsqueda del número de artículos académicos por año con el tema «Virtual Reality» se obtiene esta gráfica:

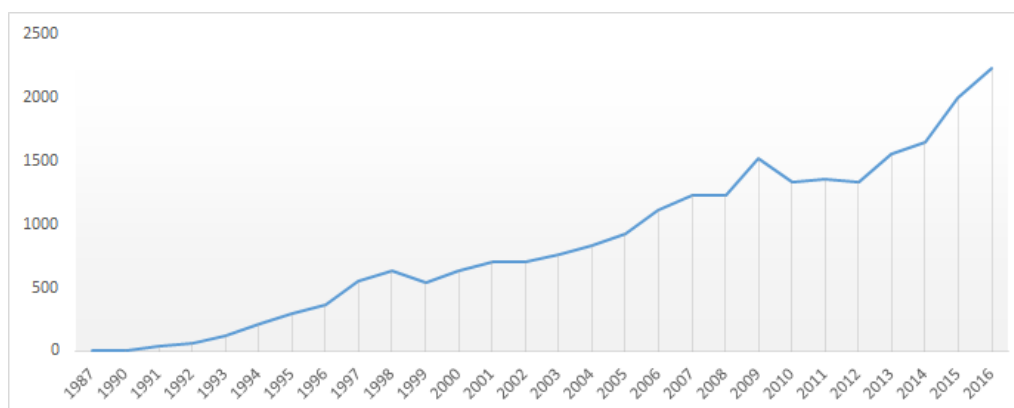


Figura 2.2: Evolución en el tiempo de las publicaciones académicas relativas a "Virtual Reality".<sup>en</sup> WebOfKnowledge

Si volvemos al gráfico anterior, aparte del crecimiento enorme a partir de 1992, se observa un punto de inflexión en 1998. Tras esa fecha el concepto va mermando en el corpus literario poco a poco. Si bien es cierto que el descenso no es tan acusado como su ascenso, es innegable su descenso.

A la vista de esta información podría parecer que esta tecnología, o al menos el interés en ella, está en retroceso. Sin embargo es necesario tener en cuenta tres factores fundamentales a la hora de interpretar esta información:

- La información del corpus de Google Books Ngram Viewer sólo llega hasta 2008.
- La citada herramienta es un recurso valiosísimo a la hora de investigar la evolución de los términos en el tiempo a largo plazo sobre todo en la era pre-internet cuando la información estaba casi en su totalidad en los libros. En la era actual este precepto pierde validez.
- El término «*realidad virtual*» es en sí mismo confuso y vago y aunque haya sido el germen del concepto, se han ido extendiendo sus matices (este tema será discutido más adelante)

Con la intención de mitigar las dos limitaciones indicadas vamos a utilizar otra herramienta de Google que permite examinar el interés global ante determinados términos: Google Trends. Aunque esta herramienta no distingue el interés de la búsqueda (científico, recreativo, negocios, etc.) permite indicar si un concepto sigue vivo en el «zeitgeist».

Si lanzamos la búsqueda de «virtual reality» en Google Trends nos aparece esta gráfica: <https://www.google.es/trends/explore?date=all&q=virtual%20reality>



Figura 2.3: Evolución «Virtual Reality» en Google Trends

Curiosamente, en Google Trends no se pueden realizar búsquedas antes de 2004 ya que la herramienta no conserva registros más antiguos. Sin embargo ambas gráficas se solapan en el periodo 2004-2008 y aunque no es posible compararlas directamente ya que en la primera se miden los ngramas que aparecen en los textos y en la segunda se mide el número de búsquedas normalizadas, podríamos aventurar las siguientes conclusiones de esta comparación:

- Ambas gráficas son coherentes
- Durante el periodo de solapamiento las pendientes de las gráficas, así como la morfología de la curva que describe son similares
- Es posible por lo tanto aventurar una gráfica anexa basándonos en las conjeturas anteriores

Esta es la gráfica resultante:



Figura 2.4: Interés estimativo en «Virtual Reality»

Si bien hemos de recalcar una vez más que esta gráfica no tiene validez métrica cuantitativa, podemos inferir de manera cualitativa que el interés general en la tecnología conocida como «realidad virtual» ha seguido la siguiente línea temporal:

- 1985-1990: Comienzos
- 1990-1996: Aumenta el interés de manera más que considerable ( $\delta > 1$ )
- 1996-1998: El interés crece menos cada vez ( $\delta < 0$ )
- 1998-2012: El interés decrece más o menos lentamente ( $\delta < 0$ )
- 2012-2014: El interés se mantiene estable ( $\delta \approx 0$ )
- 2014-2017: Vuelve a existir un aumento casi exponencial

En resumen, a principios de 2017 la realidad virtual:

- Nació hace unos 30 años
- Tuvo un gran boom a los 12 años
- Tras esto, ha estado 16 años estancado
- En los últimos 2 años vuelve a ser un valor en alza

Por último, para poner en contexto la relevancia de la realidad virtual frente a otros temas actuales, sirva esta comparación también realizada con Google Trends entre los términos «virtual reality», «streaming» e «iphone»:

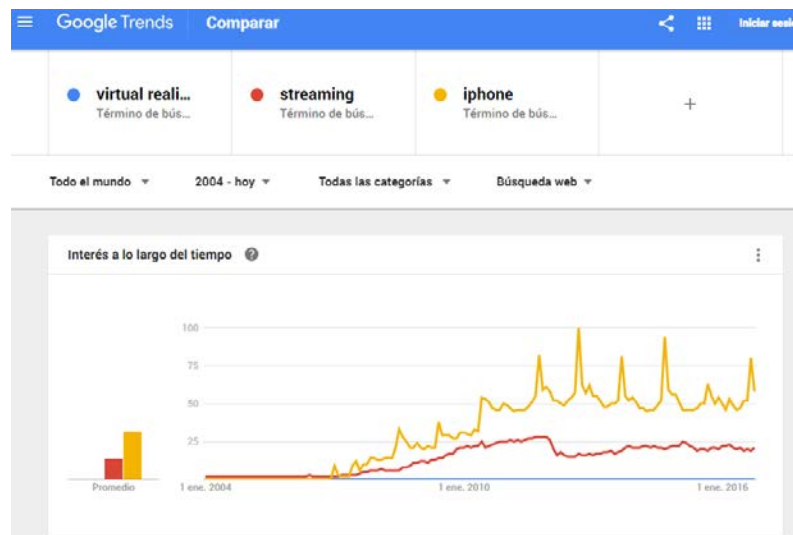


Figura 2.5: Interés comparativo «Virtual Reality»

<https://www.google.es/trends/explore?date=all&q=virtual%20reality,streaming,iphone>

Como puede observarse, el interés relativo de la realidad virtual es bajísimo respecto de otros tecnológicos o comerciales.

La conclusión inmediata de esta información es que la realidad virtual en todas sus facetas (tecnología, mercado, contenidos digitales, metodología, etc.) es todavía muy inmadura.



### 2.2.3 Surgimiento, caída y resurgimiento

En el artículo de la revista digital «The Verge» - «The Rise and Fall and Rise of Virtual Reality» (Katie Drummond and Zelenko., 2014) se realiza una retrospectiva completa de lo que ha significado la realidad virtual en el entorno tecnológico, artístico, económico y académico a la vez que se sugieren cuales pudieron ser los factores desencadenantes del gran letargo entre el fin del milenio y 2014.

- Hardware inmaduro: Aunque las primeras versiones de cascos **V.R** datan de los años 70 e incluso en los 50 y 60 se realizaron intentos un tanto pueriles de dotar a sus usuarios de cierta sensación «inmersiva», fue un fracaso. Peso excesivo, fragilidad, poca resolución o tamaño exagerado fueron algunos de los apelativos que se ganaron los dispositivos de la época. La escasa potencia gráfica y de cálculo en tiempo real del hardware gráfico tampoco ayudaron a la experiencia.
- Internet, 2.0, red social, dispositivos móviles... : La eclosión de Internet y el comienzo de su uso masivo coincide con el declive de la **V.R**. Son tecnologías complementarias, obviamente, pero el interés económico y social fue totalmente enfocado otro sentido. Tras Internet, una serie de olas derivadas del uso de la anterior ha ido recogiendo todo el interés, dejando la **V.R** en un plano muy inferior.

Parece que queda probado que hasta 2014 no existió un interés auténtico por la **V.R**. Pero ¿qué ocurrió en este año para cambiar tan bruscamente la tendencia imperante en los 15 años anteriores?

- En dicho artículo se indica que esta fecha coincide con la compra de la compañía «Oculus Rift» por el gigante «Facebook» por 2.000 millones de dólares.
- También en 2014 Markus Persson vende su empresa «Mojang AB» y su popular juego 3D «Minecraft», a Microsoft por 2500 millones de dólares.
- En el Google I/O (una conferencia de desarrolladores anual de Google) de 2014 se regaló a todos los asistentes la primera versión de las «Google Cardboard», una suerte de gafas de **V.R** muy básicas construidas con cartón, unas lentes de resina y unos imanes de muy bajo costo que utiliza el teléfono móvil como base computacional en lugar de un PC de gama alta como es el caso de Oculus Rift.

Pero el simple hecho de que importantes empresas del sector apuesten fuertemente por cierta tecnología no es en sí mismo suficiente. En el artículo de The Verge se argumenta que dichas decisiones empresariales, si bien actuaron como catalizador, son consecuencias lógicas de la paulatina eliminación de los dos factores que estaban frenando el avance de la V.R.

### 2.2.4 Breve historia comparativa

Internet sigue siendo un valor fundamental y creciente en las vidas de una parte de la población de la tierra hoy en día, pero tras una primera fase de mayor impacto, el uso de la red se ha ido normalizando día a día. Curiosamente la misma palabra «Internet» está quedando en desuso y muy ligada a los dispositivos clásicos como los PC de sobremesa, dando paso a otros como «4g», «edge» o simplemente «wifi» más propios del mundo «Smartphone». <https://www.google.com/trends/explore?date=all&>

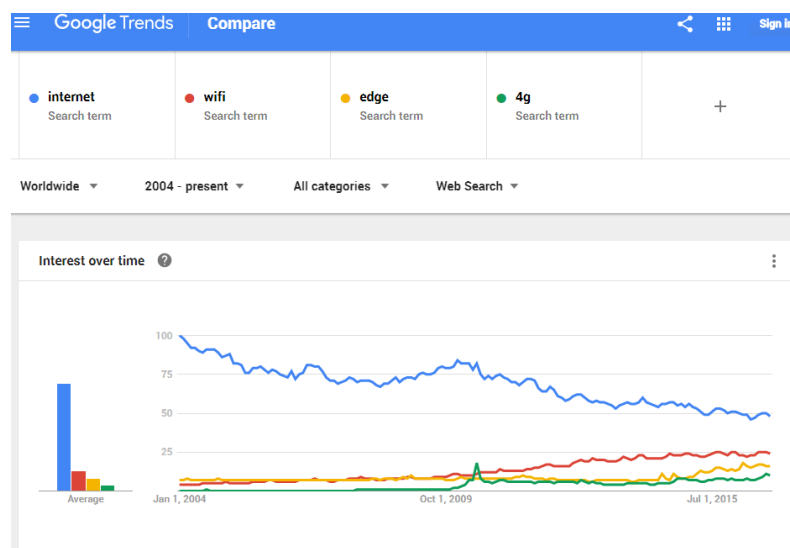


Figura 2.6: Evolución del interés en distintos términos equivalentes

q=internet,wifi,edge,4g

Es preciso hacer hincapié en que en absoluto se argumenta que el interés en Internet decaiga, todo lo contrario. Simplemente se está cambiando la forma de denominar, vivir y entender el concepto como un servicio básico de la sociedad. Cabe entonces cuestionarse si la Realidad Virtual tal y como se concibió en los años 80 tenía sentido sin contar con Internet como la entendemos hoy.

Pero puede que el mayor freno de la **V.R** durante los últimos 15 años fuese la pobre experiencia inmersiva que ofrecía el hardware, y este factor sí que está cambiando de manera drástica en los últimos años. Si bien es cierto que equipos de alta calidad como Oculus Rift, Vive o Gear VR requieren de una gran inversión y las soluciones más económicas como Google Cardboard dejan mucho que desear, están a años luz de los pesados y aparatosos cascos VR de los años 90.

En resumen, el concepto de VR nace en los años 50 del siglo XX, comienza su andadura más o menos como la conocemos hoy en los 80, tiene una edad dorada en los 90 y cae casi en el olvido hasta 2014. Estamos tratando por tanto con una idea cuya entidad va más allá de un simple avance tecnológico y se convierte en una nueva manera de entender la experimentación humana que trasciende lo que existe hasta la fecha.

Es imposible saber a ciencia cierta si la **V.R** será algo que cambie el mundo tal y como lo conocemos al igual que ha hecho internet o la tecnología móvil. Por ejemplo todavía existen problemas importantes que resolver como el «cybermareo» (Ruddle, 2004) que podrían detener la penetración de la tecnología en la sociedad.

En cualquier caso el creciente interés por la materia es innegable, las millonarias inversiones en la tecnología siguen sucediéndose e independientemente de la aceptación mayoritaria por la sociedad ya existe una masa crítica (Julier et al., 2015) que justifica un proceso industrial y asegura un mínimo de ciclo de negocio.

### 2.2.5 Usos prácticos de la Realidad Virtual

Una vez hemos fijado el interés tanto socioeconómico como académico en la Realidad Virtual debemos profundizar en su naturaleza y en sus características para comprender su alcance en los distintos campos, sectores o industrias donde puede ser aplicada.

Es definida por ( <http://www.dictionary.com/browse/virtual--reality> ) como *una simulación inmersiva y realista de un entorno tridimensional creado utilizando software interactivo y hardware, y experimentado o controlado por el movimiento del cuerpo* y por (Pimentel and Teixeira, 1993) citado por (Tötösy de Zepetnek and Sywenky, 2010) como *una experiencia inmersiva e interactiva generada por un ordenador*. En base a esta amplia definición es extremadamente complicado acotar los usos de la Realidad Virtual a un conjunto discreto de sectores industriales. Con el objetivo de poder realizar una matriz cruzada de sectores industriales y posibles usos de la tecnología **V.R** vamos a

recopilar las industrias existentes, tarea que tampoco es fácil dada la falta de consenso global al respecto.

### 2.2.6 Sectores e Industrias existentes

Según la revista Forbes (<http://www.forbes.com/sites/liyanchen/2015/09/23/the-most-profitable-industries-in-2015/#1f1db6a97712>) las industrias más importantes según el nivel de beneficios de 2015 son las siguientes por orden descendente:

Tecnología de la Salud	Financiera	Servicios Tecnológicos	Tecnología Eléctrica
Bienes de Consumo no duraderos	Transporte	Servicios de Consumo	

Tabla 2.1: Sectores Importantes por Forbes

Según la Organización Internacional del Trabajo OIT (<http://www.ilo.org/global/industries-and-sectors/lang--es/index.htm>) que es una agencia especializada de las Naciones Unidas existen los siguientes sectores e industrias:

Agricultura; plantaciones, otros sectores rurales	Alimentación; bebidas; tabaco	Comercio	Construcción
Educación	Fabricación de material de transporte	Función pública	Hotelería, restauración, turismo
Industrias químicas	Ingeniería mecánica y eléctrica	Medios de comunicación; cultura; gráficos	Minería (carbón, otra minería)
Petróleo y producción de gas; refinación de petróleo	Producción de metales básicos	Servicios de correos y de telecomunicaciones	Servicios de salud
Servicios financieros; servicios profesionales	Servicios públicos (agua; gas; electricidad)	Silvicultura; madera; celulosa; papel	Textiles; vestido; cuero; calzado
Transporte (incluyendo aviación civil; ferrocarriles; transporte por carretera)	Transporte marítimo; puertos; pesca; transporte interior		

Tabla 2.2: Sectores Importantes por según OIT

Por otra parte, el Portal Europeo para las PYMES ([http://ec.europa.eu/small-business/policy-statistics/economic-sectors/index\\_es.htm](http://ec.europa.eu/small-business/policy-statistics/economic-sectors/index_es.htm)) define los siguientes sectores económicos:

Actividades rurales	Agricultura	Aparatos de gas	Audiovisual y de medios
Biotechnología	Calzado	Comunicaciones	Construcción
Cosméticos	Cuero	Cultura	Industrias de la defensa
Economía social	Energía	Equipos a presión	Equipos radioeléctricos y equipos terminales de telecomunicación
Espacio	Fabricación	Industria alimentaria	Industria automovilística
Industria farmacéutica	Industria mecánica	Industria textil y de la confección	Industrias aeronáuticas
Industrias forestales: madera, pasta y papel, impresión	Industrias marítimas	Ingeniería eléctrica	Instrumentos de medida
Juguetes	Farmacéuticas	Minería, metales y minerales	Mueble
Pesca	Productos sanitarios	Productos pre envasados	Productos químicos
Seguridad	Seguridad alimentaria	Servicios	Servicios a empresas
Servicios al por menor	Servicios financieros	Sectores de la Sociedad de la Información	Transporte
Turismo			

Tabla 2.3: Sectores según Portal Europeo PYMES

Como se puede constatar no existe un criterio unificado internacional (o local) sobre cuáles son los sectores e industrias existentes en la actualidad. Sin embargo sí que se puede cruzar la información de las distintas fuentes disponibles y extraer, al menos, las más prometedoras.

Alimentación y Bebidas	Automoción	Construcción
Distribución y Retail	Educación	Electrodomésticos
Energía	Entretenimiento	Farmacéutico
Financiero	Hostelería	Industria
Instituciones y ONGs	Marketing	Medicina
Medios de Comunicación	Militar	Minería
Moda y Complementos	Neurociencia	Neurosicología
Ocio y Cultura	Químico	Robótica
Seguros	Sicología	Tabaco
Tecnología	Telecomunicaciones	Transporte
Turismo	Urbanismo	

Tabla 2.4: Sectores a estudio

Escogeremos este conjunto de **32** industrias como base comparativa.

### 2.2.7 Campos de aplicación de la Realidad Virtual

Existe un número astronómico de publicaciones sobre los usos de la realidad virtual aplicada a los diferentes sectores. Tanto es así que enumerar exhaustivamente todas las publicaciones que se han realizado hasta la fecha en todos los sectores y campos de aplicación escapa del alcance de la presente publicación. Sin embargo se ha hecho un esfuerzo por resumir y resaltar los estudios más importantes o ilustrativos de cada campo.

#### 2.2.7.1 Automoción

**En (Berg and Vance, 2016) los autores** parten de un punto inicial en 1999 cuando Fred Brooks, al publicar sobre el estado del arte de la **V.R** (Brooks, 1999) dijo «*La V.R ha llegado, pero a penas funciona*». Más de 15 años después, quieren revisar el estado del arte en la industria de la automoción para ver si ha cambiado algo. Para ello se embarcan en el ambicioso proyecto desde finales de 2014 hasta principios de 2015 en la que realizan visitas en persona a 18 fábricas y factorías de la industria de la automoción donde se utiliza la **V.R** en el proceso de toma de decisión en fase de diseño y en ingeniería de negocio.

Los autores entrevistan a 62 personas de varias disciplinas con distintas perspectivas para poder realizar una encuesta formal y encontrar las historias de éxito, los puntos fuertes así como los problemas y puntos de mejora de la tecnología que usan en su entorno laboral cada día. Durante su estudio recaban información en las más importantes empresas de automoción estadounidenses como *Ford, TACOM, Lockheed Martin, NREL, INL, John Deere, Case New Holland*, etc. Como conclusiones, la primera y más importante es que a diferencia de 1999, la **V.R** *!funciona!* en 2017.

Queda comprobado que los procesos industriales son mejorados a través de las herramientas **V.R**. Pero también se recopilan los puntos de mejora que los usuarios reclaman a la tecnología: reducir la latencia (mareo), render de modelos más grandes, retroalimentación háptica más satisfactoria, mayor interacción con mundos virtuales, mejoras en la modelación 3d, etc. En general todas las peticiones van en dos sentidos: mejorar la experiencia de usuario en términos de hardware y mejorar el realismo de la **V.R** dotándola de contexto espacial.

En resumen, la industria de la automoción usa la **V.R** de manera activa y quiere aumentar su uso, para lo que requiere mejoras en la tecnología tanto hardware como



software.

**En (Stork et al., 2012) se estudia** la experimentación de una cadena de montaje de automóviles que evalúa la sustitución del entrenamiento clásico por un entrenamiento en **V.R.** Los sujetos del experimento son los trabajadores de la cadena de montaje que bien por razones de reciclaje o por su falta de veteranía requieren practicar la memoria muscular de las rutinas que se requieren para operar en la cadena de montaje.

Aunque pueda parecer un contrasentido, ya que en el campo de la robótica industrial las cadenas de montaje llevan siendo un éxito desde hace décadas, cada día es más necesaria la intervención humana en las cadenas de montaje de automóviles. Este hecho se debe a la gran variedad de acabados de un mismo modelo, con variaciones de motorización, habitáculo, electrónica, seguridad, etc. necesarias para mantener la competitividad en el sector.

Como resultado de dicha competitividad, los fabricantes se ven obligados a acortar al máximo el *Time-to-Market* lo que impacta directamente en cambios constantes en los procedimientos productivos y de la cadena de montaje. Dado que los elementos de robótica industrial son lentos y caros de cambiar existe una clara tendencia en el sector para reintroducir operadores humanos que sean capaces de producir las personalizaciones del producto final que se requieran mucho más dinámicamente. Sin embargo los equipos de trabajadores humanos, a diferencia de los robots, deben ser entrenados para poder trabajar de manera efectiva y coordinada.

Es en este punto donde este estudio entronca, intentando reducir los costes y los tiempos de formación al personal operario, sobre todo evitando la construcción de instalaciones de entrenamiento que son costosas y solo pueden ser utilizadas por un alumno en cada caso. La idea de poder usar **V.R** posee muchas ventajas ya que reduce costes y acorta el tiempo de formación de los alumnos.

Existen dos grandes obstáculos para el éxito del experimento: el primero, la dificultad intrínseca de construir entornos virtuales, sobre todo si han de contar con características físicas y geométricas muy exactas. El segundo, pero no menos importante, consiste en la falta de aceptación e integración por parte de los sujetos del experimento. Los trabajadores se resistían a este tipo de formación virtual por desconocimiento o falta de confianza en el método.

Para evitar ambos obstáculos los autores del estudio proponen la utilización de una

plataforma virtual *gamificada* que haga uso de estándares y sea capaz de integrar distintas fuentes de información geométrica y espacial.

### 2.2.7.2 Educación

**En (Kaufmann et al., 2000) los autores** se centran en las experiencias con la herramienta *Studierstube* <http://studierstube.icg.tugraz.at/main.php> (Szalavári et al., 1998) (Schmalstieg et al., 1996) (Fuhrmann and Schmalstieg, 1999).

Esta es una de las primeras y más importantes experiencias construyendo herramientas **V.R** con fines docentes. El objetivo es facilitar a los alumnos la comprensión y visualización geométrica espacial avanzada, es decir, cómo se expresan en el espacio 3D las funciones matemáticas con las que están trabajando de manera abstracta. Los autores se apoyan en numerosos trabajos previos en los que se demuestra la gran ventaja cognitiva que se desprende de poder percibir directamente las formas y geometrías en 3D. Con este objetivo, y ayudado de hardware especializado, unas gafas mixtas de Realidad Aumentada y una suerte de ratón 3D primigenio, la herramienta permite la representación espacial de objetos simples en el espacio y la interacción de los alumnos con dichos objetos.

Se formalizó un experimento con 14 alumnos de la universidad de Viena de perfiles variados. Dicho experimento consistía en dos partes: en la primera los alumnos interactuaban con la herramienta para resolver un problema geométrico de complejidad media, y más tarde cumplimentaban un formulario para recibir feedback del experimento.

El resultado fue bastante positivo ya que los alumnos mostraron una rápida comprensión del interfaz de usuario y pudieron resolver el problema en su totalidad. En el formulario se constató que todos evaluaron como mucho más sencillo comprender el problema a través de la herramienta 3D que en un soporte 2D como papel o pantalla. Por otro lado seis de los estudiantes se marearon, lo que viene denominándose comúnmente *cybersickness*.

**En (Rizzo et al., 2006) los autores** presentan el uso de un aula virtual para tratar los casos de niños que presentan TDAH (*Trastorno por Déficit de Atención e Hiperactividad*). En el estudio se indica que de manera tradicional el tratamiento de estas enfermedades se basa en la observación de los pacientes en un entorno funcional en el mundo real. Dicha observación se complementa con psicometrías tipo *papel y lápiz* o como mucho

con software típico de ayuda a la toma de datos. Una vez se evalúan las observaciones se pueden diagnosticar las pautas y rehabilitar los comportamientos.

El uso de **entornos virtuales** permite recrear los entornos funcionales de una manera mucho más flexible y rápida. Por ejemplo, para simular un aula con alumnos es necesario una habitación controlada y un número indeterminado de actores en el papel de compañeros. Además si se han de recrear situaciones que disparan el trastorno, el método clásico es muchísimo más lento. Mediante el uso de la **V.R** es posible exponer al paciente a situaciones complejas, observar directamente el comportamiento, ejecutar una técnica de rehabilitación inmediatamente y volver a someter al paciente a la misma situación para poder observar la efectividad de la técnica. Otra ventaja de la **V.R** es que en ocasiones es directamente imposible utilizar un entorno concreto bien por su peligrosidad inherente (al volante de un automóvil) o bien por su naturaleza (un avión en vuelo).

Antes de la existencia de la **V.R** la construcción de estos entornos funcionales complejos se realizaba mediante *maquetas* o escenarios donde se llegaba a un consenso de realidad. Sin embargo una rama completa de opinión dentro de la psicología cuestionaban la **validez ecológica**, es decir, cómo de válidas son las terapias realizadas en entornos no-reales ya sean en **V.R** o en maquetas. Aunque esta visión sigue existiendo hoy en día, las mejoras continuas de la **V.R** van haciendo que esta opción gane adeptos.

En el caso concreto de este estudio se desarrolló un **Aula Virtual**, donde parámetros como el nivel de distracción o de estímulos focalizantes pueden ser modificados por los terapeutas. Bajo estas condiciones, los niños con TDAH son estudiados en los distintos parámetros del aula virtual pudiendo recopilar datos directamente del hardware sensorial que debe utilizar el paciente, como por ejemplo *eye-tracking*.

Aparte de las conclusiones clínicas del estudio, los autores aseveran que el uso de herramientas **V.R** permiten mucha mayor efectividad del tiempo dedicado en los pacientes, así como un abaratamiento del coste de los tratamientos.

**Es interesante el estudio de (Bricken and Byrne, 1993)** ya que se llevó a cabo en 1993, durante la primera época de auge de la **V.R**. El trabajo relata un experimento que se llevó a cabo durante un campamento de verano con un grupo de estudiantes de hasta 18 años. Se dotó a los estudiantes de unos equipos de **V.R** completos y un software de inmersión y colaboración. Se les asignó la tarea de construir entornos virtuales a su gusto por equipos y posteriormente evaluar los del resto de equipos. Aunque la validez

científica del experimento no es demasiado elevada, en este caso podemos resaltar que hace casi 25 años que se llevan haciendo experimentaciones relacionadas con la **V.R** y la educación con un éxito notable, incluso a pesar de la pobre experiencia de usuario de los equipos de la época.

#### 2.2.7.3 Entretenimiento

**En (Kade et al., 2015) se propone** un sistema de Realidad Mixta para la asistencia a los actores que están interpretando escenas de captura de movimiento en escenarios virtuales. En distintas publicaciones anteriores se han descrito las dificultades que, incluso actores con amplia experiencia, tienen para imbuirse en el papel cuando en lugar de escenario tienen un simple croma verde. Con el éxito y la consiguiente proliferación de los *CGI* en la industria del cine y del entretenimiento esta situación es cada vez más común, requiriendo investigación y soluciones al problema.

Según este estudio las gafas tradicionales de **V.R** (como Oculus Rift por ejemplo) podrían entorpecer sus movimientos y su percepción del entorno. Para evitarlo proponen un sistema basado en un teléfono móvil, un pico proyector y una batería unida al resto por una cinta elástica que se coloca sobre la frente del actor. El pico proyector lanzaría una imagen generada por el teléfono móvil justo delante de los ojos del actor con la idea de darle una referencia del entorno final de la producción.

Aunque no queda debidamente demostrado la premisa de que este sistema sea realmente mejor que unas gafas **V.R** al uso y los problemas derivados de portar un picoreflexor en la frente no son claramente explicados, el experimento sirve para ilustrar la problemática que existe en la industria del cine y el entretenimiento en general ante el rápido avance de los *CGI* y la captura de movimiento de los actores.

#### 2.2.7.4 Farmacéuticas

**En (Rothbaum et al., 2014) se presenta** un estudio clínico doble ciego para la evaluación de dos fármacos para el tratamiento del trastorno de **estrés postraumático (TEPT)**. En este caso se utiliza la **V.R** como terapia probada para el tratamiento no químico del *TEPT* y potenciador de los fármacos, lo que hará más sencillo trazar su efectividad.

Aparte de los resultados clínicos, en la experimentación se describe detalladamente cómo se utilizó la **V.R** como tratamiento del *TEPT*. El **nivel de inmersión que se requiere es extremo** y se utilizaron unas gafas de **V.R** de alto rendimiento junto con estimulación externa. En conjunto los sujetos del experimento eran estimulados mediante la vista, el oído el olfato y el tacto. A través de las gafas **V.R** se presentaba a los pacientes en una representación generada por ordenador (*CGI*) de un paisaje de Oriente Medio como Iraq o Afganistán con feedback de movimiento. Aparte de los sonidos naturales del entorno se añadían elementos de estrés como sirenas, explosiones o gritos. Olfativamente se generaban aromas a neumático quemado o a pólvora. Utilizando un subwoofer de graves se dotaba de vibraciones a todo el set de pruebas. Como resultado integrado los pacientes eran sometidos a sesiones de 45 a 90 minutos con un grado de **inmersión** altísimo. Así mismo se sometía a los pacientes a mediciones frecuentes del nivel de cortisol para medir el grado de pánico al que se estaban sometiendo.

De esta experiencia se desprende la importancia del **nivel de inmersión** para que la **validez ecológica** de un tratamiento sea aceptable.

#### 2.2.7.5 Industria

**En (Mujber et al., 2004) los autores** realizan un estudio teórico sobre las posibles aplicaciones de la **V.R** en los procesos de fabricación industrial. Tras una breve justificación del potencial de la **V.R** como herramienta básica de visualización y control de la *manufactura virtual*, se definen las aplicaciones posibles de la misma: diseño, gestión de las operaciones y proceso de manufactura. A su vez, estas tres fases pueden subdividirse según esta categorización:

- **Diseño**
  - **Diseño:** Fomenta el trabajo colaborativo al compartir el mismo entorno y mejora la visualización y la comprensión intuitiva de la experimentación con el modelo
  - **Prototipado:** Reduce la cantidad total de prototipos físicos necesarios y mejora la toma de decisiones respecto a la elección de candidatos
- **Gestión de Operaciones**

- **Planificación:** mejora el entendimiento del plan además de permitir a los usuarios la interacción y la toma de decisión sobre el propio plan
  - **Simulación:** valida y acredita las herramientas de simulación, además de permitir al personal comprender los resultados
  - **Entrenamiento/formación:** ofrece un entorno virtual y seguro donde practicar y también ayuda a valorar decisiones
- Proceso de manufactura
    - **Mecanización:** evaluación y estudio de los factores que afectan a la calidad y costes
    - **Ensamblaje:** reduce el tiempo de diseño del ciclo de diseño
    - **Inspección:** modela y simula el proceso de inspección, así como las herramientas de inspección. Además dota de un entorno donde estudiar los métodos de inspección, detección de colisiones, planes de inspección, etc.

En resumen, casi todas las fases de la fabricación industrial pueden mejorarse con el uso de la **V.R.** Los factores comunes son la mejor comprensión de la información técnica, el ensayo de actividades en entornos complejos y la compartición de datos.

#### 2.2.7.6 Juegos y Serious Games

En (Zyda, 2005) se presenta un estudio sobre la importancia e influencia de la **V.R.** en los llamados *serious games*. El autor se embarca en la compleja tarea de definir claramente qué es un serious game, remarcando el gap generacional de aquellas personas que han nacido y crecido teniendo contactos con videojuegos y otros contenidos digitales gamificados y los que no, colocando la barrera de tiempo en la década de los 80 del siglo pasado. Una vez definido el concepto de juego, define el termino como: «*Serious Game: un concurso mental, jugado con una computadora de acuerdo con reglas específicas, que usa el entretenimiento para promover la capacitación gubernamental o empresarial, educación, salud, políticas públicas y objetivos de comunicación estratégica*».

Aunque podemos también definir el término en contraposición a los juegos clásicos añadiéndole el concepto *pedagogía*, para ilustrar dicha influencia con un caso práctico, el autor hace referencia al juego «*America's Army*» que fue presentado en 2002 y estudiado

por la «*US Army*» para evaluar su posible uso como herramienta de entrenamiento para soldados. En un principio la armada estadounidense aceptó el juego como un medio para captar nuevos reclutas pero lo descartó como medio de entrenamiento al **faltarle la inmersión suficiente**. Meses más tarde un sargento de la propia armada, a modo personal y tras reflexionar sobre los problemas de los reclutas para el disparo con rifle y los campos de obstáculos, pidió a los mismos que entrenasen usando dicho juego aun en contra de lo estipulado por sus superiores. El resultado fue que los reclutas que se completaron su entrenamiento físico con el del *serious game* mejoraron tanto sus puntuaciones que sobrepasaron al resto.

Sobre esta base el autor entronca con los avances en **V.R** y los dispositivos HW asociados que mejoran enormemente la experiencia docente en general. De hecho el autor extrapola los resultados de la armada norteamericana al simple hecho de *aprender jugando* como vuelta a los orígenes del aprendizaje. En otras palabras, el hecho de disfrutar de un entretenimiento no debe ir en detrimento del aprendizaje en la misma experiencia, y lo yuxtapone al enfoque clásico y académico de la docencia donde se elimina cualquier referencia al *juego*.

Por último el autor defiende que para que los *Serious Games* del futuro puedan gozar del reconocimiento académico deberán ser apadrinados y contruidos cuidadosamente en entornos puramente universitarios. También se citan algunas experiencias de programas educacionales como del «*Viterbi School of Engineering*» que ha formado un programa interdisciplinar llamado «*GamePipe Laboratory*» donde investigan, desarrollan y prueban las tecnologías y técnicas necesarias para crear futuros planes de estudio supereficientes basados en *serious games*.

#### 2.2.7.7 Marketing

**En (Lui et al., 2007) los autores** hacen un repaso sobre los posibles usos de la **V.R** en la industria del Marketing. Para ello toman el paradigma **cognitive fit** del mundo de la administración empresarial. Este paradigma dice que para poder realizar una toma de decisión óptima es necesario que toda la información y herramientas sobre las que nos apoyemos para ejecutarla deban ir enfocadas a esa tarea. En este caso los autores aplican este paradigma al proceso de toma de decisión que un usuario final tiene siempre a la hora de decidir comprar o no un producto en concreto.

En este sentido los autores postulan que si bien la **V.R** es un campo con un potencial

excelente para el marketing, es necesario pensar en todo momento en las características diferenciadoras de los productos que se están anunciando y si esos productos pueden ser experimentados correctamente a través de la **V.R.** Por ejemplo un vino o un perfume son difícilmente anunciables en **V.R.**

Por último citan varios ejemplos entre los que destaca la experiencia en Second Life de distintas empresas que utilizaron dicho servicio para promocionar sus productos.

**En (Tussyadiah et al., 2016) los autores** se proponen evaluar el potencial de la **V.R** como vehículo para ayudar a los consumidores a la hora de planificar y contratar sus vacaciones. Manejan una serie de conceptos psicológicos muy interesantes como el de «**telepresencia**» que definen como una ilusión perceptual de primera mano, o el de la metáfora de navegación, mediante la cual se perciben los Entornos Virtuales como un **lugar** en en contraposición de una serie de material multimedia. También se utiliza la **metáfora del transporte** que se basa en dos factores: la llegada y la partida. Al explorar los mecanismos psicológicos y mentales que caracterizan un viaje en el mundo real es posible replicarlos para conseguir una experiencia más antropomórfica.

Tras someter a distintos grupos de sujetos a una serie de exposiciones a la **V.R** mediante aplicaciones estándar se procedió a entrevistarlos poniendo especial énfasis en estos conceptos:

- **Presencia espacial:** la presencia o la experiencia en primera persona de estar en un lugar, en este caso una ciudad, es el objetivo primario de la **V.R.** Al evaluar este concepto se han observado las siguientes pautas: sensación de realidad, atracción al destino, experiencia social, llegada/salida y discontinuidad
- **Actitud y comportamiento:** todos los participantes aprendieron en mayor o menor medida sobre las ciudades que experimentaron, y la mayoría expresaron intenciones de visitar dichas ciudades para tener la *experiencia completa*.

En resumen, la experiencia fue positiva aunque existieron distintas quejas relativas a la calidad inmersiva final debidas casi siempre a detalles técnicos.



### 2.2.7.8 Medicina

En (Baldominos Gómez et al., 2015) se explora el potencial médico de la **V.R** para la rehabilitación física. En concreto se ha utilizado un conjunto de Hardware y Software **V.R** para generar un **Entorno Virtual Gamificado** en el que se requiere al paciente realizar un conjunto establecido de movimientos de brazos de abducción y aducción para lograr la rehabilitación del manguito rotador después de una lesión.

Un problema recurrente en rehabilitación y fisioterapia ocurre cuando un paciente tiene asignados una serie de ejercicios que no requieren de la supervisión constante de un fisioterapeuta. Bien por el dolor de los ejercicios o por cansancio muchos pacientes tienden a ejecutar los ejercicios menos correctamente según avanza el tiempo de ejecución. Este problema hace que la fisioterapia sea menos efectiva e incluso a veces genera lesiones secundarias.

En el presente experimento, al estar gamificado, esta corrección postural se traduce en una mejor o peor puntuación de cada ejercicio. Si la puntuación baja, el paciente obtiene feedback inmediato de su postura incorrecta e intenta corregirla.

Un interesante concepto médico que también se maneja en este estudio es la «**propiocepción**» que es la habilidad de percibir la posición de nuestro propio cuerpo y la continuidad de sus partes. Se constató que es imprescindible representar el cuerpo del paciente dentro del **E.V** para que pudiese percibir el feedback de sus acciones, al bloquear completamente la visión de sus propios miembros. Sin esta ayuda visual la agudeza de los movimientos de los pacientes decaería notablemente.

Aunque esta publicación carecía de validez como estudio clínico, ilustra perfectamente el potencial de la **V.R** en la medicina. Por un lado facilita la labor de los **fisioterapeutas** y por otro hace más llevadero el proceso de **rehabilitación** al introducir gamificación. Otra ventaja de esta tecnología radica en la posibilidad de realizar rehabilitación a distancia sin tener que desplazarse en todos los casos a una clínica, en especial cuando se trata de pacientes con dolencias graves. Como punto en contra se han detectado signos de mareo (cybersickness) debido a las conocidas carencias de calidad necesarias para una inmersión completa en **V.R**.

En (Kim et al., 2005) los autores realizan un análisis DAFO (Debilidades y Fortalezas, Amenazas y Oportunidades) también ampliamente conocido y referenciado por sus

siglas en inglés SWOT (Strengths, Weaknesses, Opportunities y Threats) a la utilización de la **V.R** como herramienta terapéutica de rehabilitación. Tras justificar el uso de este tipo de análisis, muy común en el mundo de los negocios para evaluar el uso de una herramienta, procede a mostrar los siguientes resultados:

- **Fortalezas**

- Mayor **validez ecológica**
- Control de Estímulo y Consistencia
- Retroalimentación en tiempo real
- Cuing Stimuli para apoyar «**Aprendizaje sin errores**»
- Exploración auto guiada y práctica independiente
- Modificación de la interfaz en función de las minusvalías del usuario
- Registro completo y natural del funcionamiento
- **Entorno seguro** de pruebas y entrenamiento
- **Gamificación** para mejorar la motivación
- **Entornos de bajo costo** que pueden duplicarse y distribuirse

- **Debilidades**

- El desafío de la **interfaz 1**: métodos de interacción
- El desafío de la **interfaz 2**: cables y pantallas
- Proceso de **ingeniería inmadura**
- **Compatibilidad** de la plataforma
- **Flexibilidad** Front-End
- Extracción de datos de back-end, administración, análisis, visualización
- Efectos secundarios

- **Oportunidades**

- Tecnología Emergente 1: Potencia de Procesamiento e Integración de Gráficos / Video

- Tecnología Emergente 2: Dispositivos y cables
- Tecnología Emergente 3: **Análisis** e inteligencia de **datos** en tiempo real
- Sinergia industria videojuego
- La **rehabilitación V.R** tiene buena recepción intuitiva por parte del público
- **Aceptación Académica** y Profesional
- Las comunidades Científica y Clínica estrechamente unidas con Rehabilitación **V.R**
- Integración de la **V.R** con monitorización fisiológica e imágenes cerebrales
- **Tele-rehabilitación**

- **Amenazas**

- Pocos **casos de costo/beneficio** podrían afectar adopción
- Posible foco de pleitos
- Retos éticos
- La percepción de que la **V.R** eliminará la necesidad del personal clínico
- Conciencia limitada / **Expectativas** poco realistas

Una vez realizado el análisis los autores concluyen que el uso de la **V.R** en el campo de la rehabilitación clínica está en una fase temprana, caracterizada por numerosas **pruebas de concepto satisfactorias**. Por otro lado la mayoría de las debilidades del campo están derivadas de la **inmadurez** tanto de la tecnología como de la metodología. Finalmente reclaman un lugar de prominencia en los próximos años para la **V.R** aplicada a la rehabilitación siempre que se siga invirtiendo en investigación y sobre todo si se pueden reaprovechar sinergias con otras industrias como la del videojuego.

**En (Parsons et al., 2009) se revisan** los estudios realizados hasta el momento en el campo del tratamiento y rehabilitación en pediatría. Se observan estudios realizados en campos pediátricos tan dispares como: parálisis cerebral, síndrome alcoholismo fetal, autismo, déficit de atención, etc. con distintos resultados.

Los autores recalcan el hecho de que cada tratamiento y técnica de rehabilitación ha de ser diseñada específicamente para cada patología y en algunos casos prácticamente

para cada paciente, dependiendo en gran medida el éxito de la personalización del Entorno Virtual y el interfaz de usuario (guantes, visores, etc.). En el apartado de conclusiones se hace énfasis en el **gran potencial** de la **V.R** en este campo aunque todavía es una tecnología **inmadura** y debe investigarse más en el campo.

**En (Popescu et al., 2000) los autores** describen un caso práctico de tele-rehabilitación mediante terapia física. En este experimento llevado a cabo en el año 2000 no se contaba con un entorno virtual propiamente dicho, sino que se utilizaba un prototipo de guante con retroalimentación hidráulica para requerir que el paciente practicase distintos grados de fuerza en la musculatura de la mano.

Aunque es un experimento muy limitado y ya algo anticuado tiene interés al ser una de las primeras experiencias en realizar un paciente un tratamiento médico de rehabilitación desde su domicilio utilizando técnicas muy incipientes que han ido evolucionando desde entonces. El feedback de este estudio fue muy prometedor y en su día sentó las bases de siguientes experimentos en este campo.

**En (Rizzo et al., 2000) se inicia** la investigación que ya hemos revisado en (Rizzo et al., 2006) dentro del sector educativo. Sin embargo es interesante reseñar esta publicación ya que fue una de los primeros experimentos en los cuales se hizo participar a niños en tratamiento con un entorno virtual completo. Hay que tener en cuenta que este documento data del año 2000 y los medios técnicos distaban mucho de los actuales (en 2017). Eso significa que los resultados tan prometedores que se consiguieron eran «a pesar» de la escasa calidad final de la experiencia y refuerza la utilidad de la tecnología.

**En (Satava and Jones, 1998) los autores** marcaron ya hace casi 20 años los posibles caminos que seguiría la **V.R** en el campo de la medicina. Por un lado destacaban las actividades que se estaban llevando a cabo en el campo de la rehabilitación tanto física como síquica por equipos como los del doctor Albert Rizzo en la South California University. Otra vía alternativa del uso de la **V.R** en medicina consiste en las simulaciones quirúrgicas mediante las que los cirujanos podrían practicar y planificar una intervención antes de la operación. Las acciones concretas que pueden ser entrenadas van desde las simples «basadas en aguja» como una inserción intravenosa hasta las más complejas como una laparoscopia histeroscópica.

Sin embargo para que el entrenamiento funcional de quirófano sea realmente útil es necesaria una calidad final de inmersión mucho más alta, además de mejoras en la tecnología háptica. En el momento de escribir este documento no existían ejemplos reales prácticos de entrenamientos completos con la tecnología **V.R.**, y aunque actualmente se está mejorando mucho en la tecnología, todavía no hay experiencias significativas.

### 2.2.7.9 Militar

*Se debe hacer especial énfasis en la falta de publicaciones relativas a sector militar y armamentístico. En este caso la falta de publicaciones no es indicativa de falta de interés, más al contrario. Esta situación se deriva del nivel de seguridad militar y de la habitual confidencialidad que rodea este entorno.*

**En (North Atlantic Treaty Organisation and Organisation, 2003)** la Organización para el Tratado del Atlántico Norte encargó a un grupo de investigación del RTO/NATO (Organización Tecnología e Investigación de la OTAN) un estudio con estos tres objetivos:

1. Identificar los **factores humanos** relacionados con el uso de la tecnología **V.R** para fines militares
2. Determinar el estado del **conocimiento** con respecto a esas cuestiones
3. Recomendar una agenda de investigación que abordará cuestiones críticas y permitirá producir **productos eficaces** para satisfacer las necesidades militares

El estudio comienza sentando las bases de las claves del éxito de la **V.R.**: «*La clave de la eficacia de la realidad virtual para fines militares es la interfaz hombre-máquina o la interacción humano-ordenador*». Aunque por otro lado esta premisa no hace más que reforzar la misma idea que hemos encontrado en la práctica totalidad de los estudios: el éxito de la **V.R** depende de la calidad de la experiencia inmersiva.

En el estudio se hace un repaso exhaustivo de los trabajos que los estados miembros de la OTAN han realizado de manera individual relativos a la **V.R.** Posteriormente se realiza una disertación sobre los posibles usos de la tecnología **V.R** en el ámbito militar y lo divide en los siguientes campos:

- **Evaluación de nuevos diseños de armamento:** como en cualquier otra industria los **prototipos** han de ser evaluados antes de ser construidos y la **V.R** permite acortar tiempos y costes
- **Evaluación y pruebas de Sistemas Armamentísticos:** los sistemas armamentísticos, como cualquier otro sistema, deben ser probados ampliamente. Una manera de evaluar dichos sistemas muy eficiente es **virtualizar un entorno** donde puedan realizarse toda la batería de test necesaria
- **Docencia y entrenamiento:** en todos los ámbitos existe la misma necesidad. En particular las ventajas de la **V.R** se ven potenciadas en:
  - Simuladores de Vuelo: históricamente una de las primeras aplicaciones de la **V.R**
  - Simuladores Navales
  - Simuladores de Tiro
  - Simuladores Médicos
  - Planificación y Ensayos
  - Aplicación en misiones reales: realidad virtual, mixta y aumentada
  - Visualización táctica y estratégica

Tras evaluar el estudio la Organización decidió formar un grupo permanente que promoviese y cuidase de los siguientes conceptos relativos a la **V.R**:

- Identificar las «**killer apps**»
- Identificar una lista de **objetivos, requerimientos** de usuarios y **tecnologías** para las inversiones militares
- Fomentar el desarrollo de **interfaces naturales V.R** y de **agentes** y modelos **inteligentes** con comportamiento realista
- Reunir grupos interdisciplinarios y crear vocabulario común sobre problemas compartidos
- Crear una red de **investigación** e identificar nuevas fuentes de financiación

- Compartir **bibliotecas de software** y crear un depósito central de dispositivos y módulos
- Abrir la publicación no clasificada de los resultados a otras organizaciones.

Aunque este documento no es en sí mismo una publicación científica es extremadamente ilustrativa para entender el alto grado de interés que el mundo militar (habitualmente hermético) tiene en la tecnología.

**En (Lele, 2013) se hace referencia** a (North Atlantic Treaty Organisation and Organisation, 2003) y se añade una perspectiva algo más actualizada con diez años de diferencia entre ambas publicaciones. En dicho estudio publicado fuera del ámbito militar, el autor se plantea evaluar la **V.R** en todos sus aspectos para su uso en los sistemas de seguridad de los EEUU.

Tras dar un amplio repaso desde los orígenes del concepto de realidades alternativas con Platón y la alegoría de la cueva termina haciendo un repaso de las distintas tecnologías involucradas en la **V.R**. Una de las más claras aplicaciones es el entrenamiento militar, con dos grandes ventajas: la **reducción** de exposición al **peligro** y el mantenimiento de la **confidencialidad**. Otra ventaja es la posibilidad de exponer a los soldados a situaciones inusuales que sería extremadamente caro o directamente no podrían ser replicadas de otra manera (condiciones meteorológicas extremas, mal funcionamiento de equipo, escenarios de combate inusuales, etc.).

En conclusión refuerza los mismos conceptos del estudio de la OTAN: **interés máximo sobre todo en lo relativo a simulación y entrenamiento**.

**En (Armstrong et al., 2013) se da** un enfoque del interés militar de la **V.R** desde un punto de vista médico. Los autores documentan que un gran problema médico para los ejércitos es el alto nivel de *Lesión Cerebral Traumática* (**LCT** o TBI) entre los soldados.

Estas heridas suelen ser de consideración media o leve y no suelen tener secuelas neurocognitivas, sin embargo es difícil realizar una evaluación completa. En el estudio se presenta una experimentación llamada **VRST** que consiste en presentar mediante **V.R** los estímulos más realistas posible a los pacientes de **LCT** y medir sus respuestas. De esta manera es mucho más sencillo diagnosticar y evaluar el grado de **LCT** previa a una situación de combate real.

### 2.2.7.10 Minería

**En (Wyk et al., 2009) los autores** se centran en la problemática del alto nivel de accidentes de trabajo en la industria minera de Sudáfrica. Además de promover la calidad en las instalaciones mineras, el gobierno sudafricano ha puesto énfasis en la formación y el entrenamiento en seguridad laboral.

Sentadas estas bases, el presente estudio trata de validar la validez del entrenamiento minero en **V.R.** Para ello realiza un estudio sobre una base de trabajadores teniendo en cuenta estas consideraciones:

- Los sujetos tienen diferentes orígenes culturales y hablan varios idiomas
- Algunos tienen una comprensión muy limitada del inglés
- Las edades están entre 20 y 60; Los niveles de educación varían desde el grado 5 al grado 12
- Los entrevistados tienen varios niveles de experiencia minera, que oscila entre 2 y 25 años
- Actualmente, la mayoría de los trabajadores del subsuelo son hombres, aunque algunas minas ya están nombrando mujeres en algunas de estas posiciones
- La mayoría de los entrevistados confía en que pueden desempeñar sus funciones bien
- Más del 80 % de los entrevistados nunca han utilizado un ordenador
- Para determinar el uso de la tecnología, se pidió a los trabajadores que indicasen si utilizan habitualmente teléfonos celulares y cajeros automáticos. Más del 80 % respondió afirmativamente. A pesar de las grandes diferencias en el nivel de alfabetización y la mínima experiencia con ordenadores, los sujetos no se mostraban reacios a la formación informatizada en **V.R.**

Se definieron una amplia gama de actividades formativas, unas más clásicas, con preguntas y respuestas y otras más avanzadas utilizando técnicas de **V.R.**, llegando a las siguientes conclusiones:



- Son una alternativa de **bajo costo** a la creación a gran escala de escenarios reales
- Los sistemas pueden ser diseñados para ser multilingües o directamente no necesitar un idioma
- La **V.R** es una configuración **flexible** y está abierta a modificación y personalización
- Ofrece la oportunidad de crear una gran **variedad** de **escenarios**, incluidos los raramente o nunca antes encontrados en la vida real
- Expone a los usuarios a situaciones simuladas que serían peligrosas de encontrarse en la vida real
- Esta experiencia virtual de *tiempo de trabajo* ayuda a reforzar el aprendizaje que tuvo lugar durante el aprendizaje convencional en el aula
- Los sistemas de formación de **V.R** proporcionan niveles adecuados de interacción con los alumnos
- Los capacitadores pueden revisar los registros de una extensa base de datos, comparar resultados y generar gráficos de rendimiento de individuos o grupos
- Las áreas problemáticas de los aprendices son fácilmente identificables

#### 2.2.7.11 Neurociencia

La revista *Nature* incluyó a (Tarr and Warren, 2002) en su edición de Octubre de 2002. El artículo, desde una perspectiva amigable, presentaba un paralelismo entre la **V.R** y la *Holocubierto* de *StarTrek TNG*. Más allá de la pura divulgación, este artículo presenta el concepto de percepción desde una aproximación ecológica basándose en el trabajo del eminente psicólogo perceptual J.J.Gibson en 1979. Este principio dice que *el proceso de **percepción** emerge de un organismo embebido en un entorno al **interactuar** con dicho **entorno**.*

Mediante este concepto «**ecológico**» se entronca toda la investigación neurocientífica, alabando el potencial de la **V.R** para generar un entorno virtual donde poder experimentar siguiendo este **principio de ecologismo**. En el artículo, tras presentar algunos trabajos realizados en la época, se finaliza recalcando el gran potencial de la tecnología a la espera de que el hardware y el software entren en una fase más madura.

### 2.2.7.12 Neurosicología

**En este campo son muy relevantes** los estudios de Albert Rizzo de la University of Southern California, en particular (Rizzo and Buckwalter, 1997) y (Rizzo et al., 2004), dos trabajos conectados en los que los autores definieron las bases de la **evaluación cognitiva** mediante técnicas de **V.R**. También intrínsecamente relacionada con esta evaluación cognitiva entran en el campo de la **rehabilitación** psicológica y neurofisiológica.

Sin entrar en los numerosos detalles clínicos de estos estudios y todos los que han ido desprendiéndose y evolucionando de estos, Rizzo y su equipo demuestran rigurosamente los beneficios enormes del uso de la **V.R** a la hora tanto de **diagnosticar** como de **tratar** un amplio rango de enfermedades neurofisiológicas, desde heridas traumáticas cerebrales hasta Alzheimer pasando por el síndrome de déficit de atención entre muchos otros.

Es necesario recalcar la clara intención de Rizzo et al. en demostrar la idoneidad de la **V.R** para este campo de la medicina. En concreto el primer apartado del estudio de 2004 se denomina «**WHY VIRTUAL REALITY?**» (¿Por qué realidad virtual?) y en la que enormemente enuncia el enorme potencial de la **V.R** para «*producir entornos virtuales simulados que permitan el desarrollo sistemático de desafíos cognitivos ecológicamente relevantes*». En otras palabras, el uso de la **V.R** permite a la mente humana exponerse a entornos nuevos y desafiantes que le permiten curarse o evolucionar.

### 2.2.7.13 Restauración

**En (Hwang et al., 2012) los autores** se centran en el problema del área de la restauración (hospitality) de las aglomeraciones y su impacto en el público. En concreto el estudio busca acotar la importancia de la primera impresión del público al entrar en un restaurante y observar la zona de espera. El objetivo es entender el impacto de la aglomeración existente en las emociones de los clientes y sus subsiguientes emociones de aceptación o rechazo.

Para conseguir dicho objetivo se diseñó el **entorno virtual replica** de un restaurante experimentable a través de **V.R**, navegable, foto-realístico y tridimensional. Tras ir modificando los niveles de ocupación del entorno virtual y exponiendo a los sujetos a los distintos casos de prueba se les realizaron encuestas para recibir feedback de su experiencia.

Los resultados indicaron que las respuestas emocionales eran directamente proporcionales al nivel de ocupación y que los usuarios normalmente prefieren un equilibrio entre la privacidad y la sobreocupación. Pero independientemente de los resultados este es un claro ejemplo de cómo la **V.R** puede ayudar a **tomar decisiones** de calado a los directores de cualquier tipo de establecimiento abierto al público, desde restaurantes a hoteles o salas de espectáculos.

#### 2.2.7.14 Retail

*Se debe recalcar la escasez de estudios académicos relativos al mundo del retail, aunque sí que existen numerosos artículos no académicos que reflejan el verdadero interés de la industria en la tecnología.*

**En el periódico de Internet Huffington Post** (Flanagan, 2014) se hace un breve inventario de las últimas «startups» dedicadas directa o indirectamente al mundo del retail. Aunque no existen todavía aplicaciones **V.R** completas para, por ejemplo, recorrer una tienda y probarnos ropa, existen numerosas empresas emergente que comienzan a trabajar con los fundamentos necesarios para dar ese paso (escáneres corporales, avatares inteligentes, etc.).

Por otro lado en el artículo también se recoge el **gran interés** que los grandes del sector de la moda, que por otro lado mueven enormes cantidades de dinero, tienen en todos los avances tecnológicos que impacten en su mercado objetivo, habitualmente gente joven con grandes tasas de aceptación y penetración de nuevas tecnologías.

También se recoge la vertiginosa rapidez de implantación de las nuevas tecnologías en retail: «*lo que pensábamos que tardaría diez años en ocurrir, está aquí en dos*». Con todos estos factores, tenemos un sector muy activo que está a la espera de la llegada de «*killer apps*» para poder adoptar inmediatamente la tendencia a su negocio.

**En el más reciente artículo** de la revista Forbes (Arthur, 2016) al respecto hacen referencia tanto a la **V.R** como a la **IA** de una manera general. De hecho en el texto se hace referencia a distintos cuestionarios que se han realizado a los usuarios más *early adopters* del sector retail y se constata que el gran público no distingue claramente entre ambos conceptos (**IA** y **V.R**) ya que los ven simplemente como la evolución lógica de la

tecnología. De hecho no solo estarían dispuestos a adoptar nuevos paradigmas sino que los están exigiendo.

En este artículo se citan algunos conceptos de alto nivel como *omnichannel*, *seamless commerce* o *borderless shopping* en los que se materializa este nuevo paradigma del sector retail.

Aparte de este reclamo de parte de los usuarios finales, el artículo cita algunos de los grandes *retailers* que ya están realizando pruebas de concepto en este sentido: *Tommy Hilfiger*, *Lowe's*, *The North Face*, *Lyst*, etc.

Como último apunte, se cita a Westfield (un gran centro comercial Londinense) que «predice que la VR se convertirá en ubicua en los próximos años» y a Guillaume CharnyBrunet de *Space 10* que asevera: «La realidad virtual es un tema candente en este momento y eso podría cambiar o desbaratar completamente la industria retail durante los próximos 10 o 15 años. Una vez que esta tecnología sea aceptable, todo un mundo de posibilidades se abre al retail así como a muchas otras industrias, para reinventar la forma en que solíamos hacer negocios»

**En (Guidi et al., 2010) se presenta** un caso de uso real en el que la **V.R** es utilizada por la industria *retail* para optimizar su toma de decisión. Dentro del sector existe un amplio grupo denominado «*autoservicio*» (grandes o pequeños supermercados y en general cualquier negocio en el que el cliente toma directamente los productos de las estanterías) que dedica un gran esfuerzo en el diseño y puesta en servicio de los estantes de productos.

El *time-to-market* de un cambio en esta distribución puede significar importantes pérdidas económicas en el negocio y por eso es necesario ayudarse de la tecnología existente para reducir tiempo y costes. En este sentido la **V.R** puede ayudar a la toma de decisión mediante prototipos virtuales al igual que lo hace en otras industrias que se enfrentan al mismo problema.

En el estudio, además del comentado experimento, se realiza una somera taxonomía de los posibles usos de la **V.R** a la industria retail:

- Diseño y gestión del *punto de venta*
- Investigación del mercado

- Virtual Shopping
- Configuración de productos

Excepto en el caso del *Virtual Shopping*, el resto van enfocados a ayudar en la toma de las distintas decisiones que se pueden dar en el negocio.

### 2.2.7.15 Robótica

**La relación entre el sector de** la robótica y la **V.R** es más de tipo sinérgico que de uso. En (Burdea, 1999) el autor realiza un repaso por todos los pasos que se sigue en una hipotética planta industrial robotizada desde el diseño y programación de los robots hasta la operación y mantenimiento. El autor detecta numerosas similitudes entre ambas tecnologías derivadas como es lógico tanto de su condición de *hermanas* dentro del campo de las ciencias de la computación como por su condición espacial tridimensional.

Por ejemplo ya se está realizando cierto tipo de representación gráfica lineal (CAD) que es directamente transportable a la **V.R**. Otro claro ejemplo es el feedback háptico que muchos brazos robóticos poseen y que pueden ser retroalimentados hacia guantes y actuadores hápticos en **V.R**. Esta es la tabla completa de sinergias que el autor destaca:

- Diseño CAD
  - **Tradicional:** diseños "a lápiz", sin háptica
  - **V.R:** interacción gestual manual, háptica
- Programación de robots
  - **Tradicional:** conocimiento tedioso de lenguaje robótico bajo nivel
  - **V.R:** programación a alto nivel
- Tele-operación
  - **Tradicional:** retroalimentación pobre, imposible con largos tiempos de espera, monousuario
  - **V.R:** retroalimentación mejorada, posible con largos tiempos de respuesta, multiusuarios

- Retroalimentación háptica
  - **Tradicional:** solo en algunos casos, caro y necesita expertos
  - **V.R:** interfaces específicos, seguridad y mejora el realismo de la experiencia

En conjunto se espera que la **V.R** sea capaz de ayudar a la robótica dotándola de mayor interactividad y vividez de la experiencia del usuario.

#### 2.2.7.16 Transporte

*El uso de la **V.R** en el sector transporte es inusualmente novedoso y está muy extendido. Aunque en un primer momento podría no parecer que es un sector muy proclive a su uso, la componente logística y de planificación que subyace al transporte hace entroncar la **V.R** con otras disciplinas hermanas provenientes de la **IA** como los Sistemas Multiagente y el Big Data.*

**En concreto, en (Yu et al., 2013) y posteriormente** ampliando el trabajo en (Yu et al., 2014) se centra en el modelado y simulación del comportamiento del tráfico a nivel microscópico dentro de un entorno de **V.R** utilizando **arquitectura multiagente** y **simulación distribuida**. Como en todos los campos en los que se utiliza simulación no existe un modelo matemático satisfactorio que defina correctamente el comportamiento del tráfico de vehículos. Por otro lado es absolutamente necesario prever cómo se comportará el tráfico ante actividades urbanísticas o situaciones de emergencia.

Si ya es complejo definir una simulación de tráfico habida cuenta del gran número de variables que implica, el trabajo de visualización e interpretación de resultados se vuelve prácticamente inviable. Los autores proponen un sistema multiagente para simular, en base a lógica difusa y autómatas de estados, el comportamiento de los conductores en carretera. Como conclusión relativa a la **V.R**, el estudio da por sentado esta tecnología como herramienta base de la simulación e interpretación de resultados.

**En (Erath et al., 2016) también se trata** el problema de la gestión del transporte, en este caso en una gran ciudad como Singapur. El consistorio de la ciudad está preocupado por la escasa accesibilidad peatonal y ciclista actual y está buscando definir acciones

de mejora. Los autores proponen utilizar la **V.R** en dos vertientes: ayuda a la **toma de decisiones** y **simulación humana**.

En el campo de la toma de decisiones la **V.R** es una herramienta bien conocida. En este caso tanto los arquitectos como los usuarios finales pueden tener una experiencia directa de cómo serán las posibles modificaciones urbanísticas que se realicen buscando promover el uso de la bicicleta como medio de transporte.

Por otro lado en este experimento se ha realizado una *gamificación* del entorno de tal manera que sea posible ejecutar un paseo a pie o en bicicleta por el entorno virtual de la propuesta arquitectónica. En dicho entorno virtual se han situado otros peatones, bicicletas y tráfico rodado para dar una tasa de realismo más alto. Todas las experiencias humanas en el *juego* son la base para generar resultados y extrapolar conductas y patrones de uso. Con toda esta información como retroalimentación a los arquitectos se generan nuevas versiones refinadas del modelo propuesto de manera iterativa hasta encontrar un diseño óptimo con un coste mínimo.

Como conclusión, es muy novedoso el uso de intervención humana dentro de un entorno virtual como entrada al propio sistema simulador. Aunque habitualmente este tipo de simulaciones se realiza mediante **sistemas multiagente** con un éxito notable, esta experiencia abre una nueva puerta al diseño de sistemas multiagente dando la oportunidad de utilizar «*agentes humanos*» como **generadores de patrones de comportamiento** a emular por los agentes software.

**En (Li et al., 2015) los autores** entran de lleno en el mundo del Big Data a través del concepto «*wise transportation*». Con la llegada del IoT, del cloud computing y de la mayor y mayor sensorización existente tanto en los automóviles como en las vías de tránsito en la actualidad existe una verdadera avalancha de datos proveniente del análisis del tráfico real. El tratamiento de toda esta información cae dentro del campo denominado **Big Data Analysis**. Por otro lado los autores definen «*wise transportation*» como la integración eficiente de toda la información proveniente de los sensores inteligentes, información pública de transporte y proceso de datos representados por el **car networking** y el **cloud computing** aplicado al sistema de transporte al completo. Se caracteriza como un paso evolutivo del **intelligent transportation** enfocándose más en el análisis del tráfico, descubrimiento de conocimiento y soporte a la toma de decisión.

En cuanto al uso de la **V.R** en este estudio se plantea como medio básico para la

**presentación de la información.** Al ser este un caso tipo de **Big Data** es fundamental dotar al sistema de una salida de información lo más comprensible e intuitiva posible al usuario final. La tecnología utilizada al respecto es WebVR y WebVRGIS, para conseguir analizar y predecir el tráfico urbano. El concepto representativo utilizado se basó en la cartografía del estudio, dotando de elementos de representación 3D geo localizados caracterizando distintos eventos y situaciones del tráfico.

En conclusión, es un caso que ilustra a la perfección cómo la **V.R** sirve de herramienta de **representación de datos** para la información extraída de un sistema **Big Data**.

#### 2.2.7.17 Turismo

**En (Guttentag, 2010) el autor** define seis campos de actuación del sector turístico en los cuales la **V.R** tiene potencial de uso: **planificación y gestión, marketing, entretenimiento, educación, accesibilidad y preservación de la herencia cultural**. Los cuatro primeros campos identificados son los comunes a otras áreas: **gestión del conocimiento, ayuda a la toma de decisiones, formación y entrenamiento, publicidad**, etc. Sin embargo para los conceptos de **accesibilidad y preservación del patrimonio histórico** el uso de la **V.R** es sin duda algo novedoso.

El autor desarrolla el concepto de «**recreación virtual**» como la contrapartida en **V.R** de las recreaciones en un espacio alternativo al original que ya existen en numerosas instalaciones turísticas y culturales (arte prehistórico que se degrada rápidamente al ser expuesto al público, localizaciones que por su difícil acceso hacen imposible garantizar la integridad física de los visitantes, etc.). De hecho se constata un debate abierto en estos días dentro del sector turístico que trata sobre si es posible o no que la **V.R** sea un sustituto del turismo *clásico*.

Dicho debate tiene a favor de la **V.R** los factores anteriormente citados y para casos muy concretos parece que existe consenso en que la **V.R** es «mejor que nada» en los casos en los que la experiencia directa es imposible. También es aceptable en mayor o menor medida para las experiencias monosensoriales como el examen al detalle de obras de arte (lienzos, esculturas, objetos, etc.) en los que los medios **V.R** dan mejor detalle visual/auditivo. Sin embargo distintos estudios presentan un rechazo casi total al uso de **V.R** como sustituto del turismo «de experiencia» siempre relacionado a la **falta de abanico sensorial**. En numerosas encuestas los turistas virtuales se han quejado de no poder sentir el viento en la cara (tacto/olfato), no poder degustar comida local (gusto) o



el ambiente característico de una zona concreta por ejemplo un mercadillo en el trópico (temperatura-tacto/olfato/gusto) además de no disponer de interacciones reales con personas locales.

Como conclusión el autor ve claramente el futuro del turismo en la **V.R**. Ya sea como un medio sustitutor completamente orientado a un público concreto, como un método para evaluar qué oferta es más interesante para contratar un viaje real o como medio alternativo para poder disfrutar de elementos que de otra manera estarían vetados, queda demostrado el potencial de la **V.R** en esta industria. Mirando al futuro, el estudio prevé que según la **tecnología V.R** vaya **mejorando** y abarcando mayor **abánico sensorial** mayor número de personas, sopesando pros y contras, se sentirán atraídas hacia un **turismo puramente virtual**.

**En (Huang et al., 2016) se documenta** un experimento realizado con 186 participantes con el objetivo de entender cómo los **turistas virtuales** usan un mundo virtual 3D. Los datos extraídos en dicho experimento a través de múltiples cuestionarios tanto previos como posteriores a la experiencia se estudian con distintos enfoques prácticos: el modelo de aceptación tecnológica, aproximación psicológica de la experiencia de usuario y patrones característicos orientados a los profesionales del marketing turístico.

Aparte de las conclusiones, es importante resaltar que este estudio de 2016 no estaba orientado a saber si la **V.R** es aplicable al turismo, **eso lo da por hecho**, sino que intenta **optimizar** la manera de utilizar la tecnología de la manera más efectiva posible. Es decir ya no están tratando la **V.R** como un elemento futuro sino un activo presente y además se está intentando averiguar cuáles son las **mejores prácticas y metodologías** que deben usarse.

#### 2.2.7.18 Urbanismo

**En (Nguyen et al., 2016) los autores** plantean una experiencia enfocada a ayudar a la toma de decisión dentro del campo de la planificación y construcción de ciudades (urbanismo). La propuesta se basa en saltar desde el método tradicional en el que los arquitectos diseñan por separado sus edificios casi por completo hasta las últimas fases en las que se construyen las maquetas, al nuevo método de generar un **entorno virtual inversivo multiusuario** donde se puedan ir *subiendo* los modelos prácticamente desde el momento de su concepción tanto exterior como interior. Esta posibilidad de poder

**evaluar** prácticamente desde el comienzo por parte de los usuarios (normalmente quien financia los edificios) el resultado final es una gran mejora tanto en costes, al no tener que deshacer trabajo, como en tiempos y también en satisfacción final.

Este problema no es nuevo y ya existen en el mercado numerosas herramientas tipo CAD que permiten realizar una previsualización en 3D del modelo final, pero no hay que confundir esto con **V.R** ni con la capacidad inmersiva de cara a la **toma de decisión** de los **V.I.E.**

En conclusión, los autores auguran un enorme campo de aplicación de la **V.R** dentro del urbanismo tan pronto como la tecnología **V.R** sea más barata, asequible y satisfactoria.

### 2.2.8 Sectores de con aplicación

Si tomamos de nuevo la tabla 2.4 en la que se definieron los sectores a estudio y diferenciamos sobre cuales se han publicado estudios, el resultado es el siguiente:

Alimentación y Bebidas	Automoción	Construcción
Distribución y Retail	Educación	Electrodomésticos
Energía	Entretenimiento	Farmacéutico
Financiero	Hostelería	Industria
Instituciones y ONGs	Marketing	Medicina
Medios de Comunicación	Militar	Minería
Moda y Complementos	Neurociencia	Neurosicología
Ocio y Cultura	Químico	Robótica
Seguros	Sicología	Tabaco
Tecnología	Telecomunicaciones	Transporte
Turismo	Urbanismo	

Tabla 2.5: Sectores con aplicación VR

En conclusión, de los 32 sectores generales, 24 (tres de cada cuatro, un 75 %) tienen al menos una aplicación directa de **V.R** y publicaciones tanto académicas como periodísticas que las describen. Es más, en algunos sectores como la medicina, la psicología o la industria militar existen numerosos casos (tanto de estudio como de uso) bien documentados y con experiencia más que contrastada.

### 2.2.9 Casos de Uso identificados

Hemos repasado la evolución y los campos de aplicación de la tecnología, demostrando tanto el interés global como sus enormes posibilidades prácticas presentes y sobre todo futuras.

Aunque los sectores analizados son muy dispares es posible realizar una lectura transversal de todos los estudios. Al hacerlo surgen una serie de «lugares comunes», donde curiosamente coinciden la mayoría de estudios. Para poder plasmar estas características generales se ha elaborado una matriz con todos los trabajos revisados y todos los casos de uso encontrados de tal manera que podamos visualizar mejor la información y también generar estadísticas.

Se han estudiado un total de 35 publicaciones y se han encontrado los siguientes 23 casos de uso diferenciados:

- agentes humanos
- diagnóstico
- diseño de materiales
- ensayo acciones físicas/mentales
- entrenamiento/training
- evaluación prototipos
- experimentación ecológica
- formación/learning
- lúdico
- mejora comprensión/interpretación resultados
- mejora inmersión
- mejora sensorial
- rehabilitación/telerehabilitación
- simulación
- sinergia tecnológica
- teleoperación
- toma decision
- tratamiento
- turismo virtual
- virtual shopping
- visualización big data
- visualización datos
- visualización datos grupal/data sharing

Antes de entrar a evaluar los resultados es importante remarcar que la siguiente tabla expresa únicamente el contenido de los trabajos estudiados, no el potencial completo que entraría en el campo especulativo y queda fuera del alcance de esta tesis.

## 2.2.9.1 Matriz publicaciones / casos de uso

En la presente matriz se muestran las relaciones entre las publicaciones y los casos de uso. En la primera y segunda columna se indica el sector y la publicación, estando ordenadas todas las filas por sector. En la columna tercera y en adelante se consignan todos los casos de uso encontrados. En la intersección entre publicaciones y casos de uso se coloca una «X» si en dicha publicación se describe o se ha utilizado el caso de uso correspondiente.

Sector	Autor	agentes humanos diagnóstico	diseño de materiales	ensayo acciones físicas/mentales	entrenamiento/training	evaluación prototipos	experimentación ecológica	formación/learning	lúdico	mejora	mejora inmersión	mejora sensorial	rehabilitación/tele-rehabilitación	simulación	sinergia tecnológica	teleoperación	toma de decisión	tratamiento	turismo virtual	virtual shopping	visualización big data	visualización datos	visualización datos grupal/data
AUTO	Berg2016		X		X	X											X						
AUTO	Stork2012				X	X	X																
EDU	Kaufmann2000								X	X													X
EDU	Rizzo2006		X		X	X							X					X					
ENTERTAIN	Kade2015	X					X		X		X												
FAR	Rothbaum2014																	X					
INDUS	Mujber2004			X	X	X	X				X												X
JUE	Zyda2005				X	X			X	X													
MARK	Liu2007											X											X
MARK	Tussyadiah2016	X															X						
MED	Baldominos2015									X			X										
MED	Kim2005		X		X		X		X	X			X										
MED	Parsons2009		X						X				X					X					
MED	Popescu2000												X										
MED	Rizzo2001		X										X										
MED	Satava1998					X															X	X	
MIL	NATO2003				X	X		X				X		X	X	X	X					X	
MIL	Armstrong2013		X										X										
MIL	Lele2013				X	X								X			X						
MIN	VanWyk2009				X			X															
NEUPSI	Rizzo1997		X							X			X					X					
NEUPSI	Rizzo2004		X							X			X					X					
NEURO	Tarr2002						X															X	
REST	Hwang2012					X											X						
RETA	Flanagan2014					X											X						
RETA	Forbes2016					X											X						
RETA	Guidi2010					X											X		X				
ROBO	Burdea1999			X								X			X	X							
TRANS	Erath2016									X				X			X						
TRANS	Li2015		X			X			X	X				X			X						
TRANS	Yu2013																				X		X
TRANS	Yu2014																				X		X
TURI	Guttentag2010																X		X				
TURI	Huang2016																X		X	X			
URBA	Nguyen2016			X		X											X						

Figura 2.7: Matriz de publicaciones y casos de uso

Tras observar los resultados se constató que existe cierto grado de clusterización vertical. Teniendo en cuenta que las publicaciones están ordenadas por sector es lógico que encontremos estos agrupamientos verticales significando que en los mismos sectores se realizan los mismos casos de uso.

También se observaron ciertos patrones de casos de uso (filas), aunque dado que los casos de uso están ordenados alfabéticamente es algo más complicado de observar. Este hecho parece indicar también una relación entre grupos de casos de uso lo que podría establecer una taxonomía de orden superior. Con el objetivo de observar esta posibilidad se han agrupado los casos de uso en función de dichos patrones que arrojan los siguientes resultados:

<ul style="list-style-type: none"> <li>• agentes humanos</li> <li>• lúdico</li> <li>• mejora inmersión</li> <li>• turismo virtual</li> <li>• virtual shopping</li> </ul>	<ul style="list-style-type: none"> <li>• ensayo acciones físicas / mentales</li> <li>• entrenamiento / training</li> <li>• experimentación ecológica</li> <li>• formación / learning</li> </ul>	<ul style="list-style-type: none"> <li>• diagnóstico</li> <li>• mejora sensorial</li> <li>• tratamiento</li> <li>• rehabilitación / telerehabilitación</li> </ul>
<ul style="list-style-type: none"> <li>• diseño de materiales</li> <li>• simulación</li> <li>• sinergia tecnológica</li> <li>• teleoperación</li> </ul>	<ul style="list-style-type: none"> <li>• evaluación prototipos</li> <li>• mejora comprensión / interpretación resultados</li> <li>• toma decision</li> <li>• visualización big data</li> <li>• visualización datos</li> <li>• visualización datos grupal / data sharing</li> </ul>	

Tabla 2.6: Agrupación de casos de uso por categorías

Se definen 5 categorías de tipo superior. Si plasmamos esta ordenación en la matriz anterior añadiéndole un código de color para su mejor comprensión tendremos la siguiente matriz clusterizada:

Sector	Autor	agentes humanos lúdico mejora inmersión turismo virtual virtual shopping	ensayo acciones físicas/mentales entrenamiento/training experimentación ecológica formación/learning	diagnóstico mejora sensorial tratamiento rehabilitación/tele-rehabilitación	diseño de materiales simulación sinergia tecnológica teleoperación	evaluación prototipos mejora toma decisión visualización big data visualización datos visualización datos grupal/data
AUTO	Berg2016		X		X	X
AUTO	Stork2012		X			X
EDU	Kaufmann2000			X		X
EDU	Rizzo2006		X	X	X	
ENTERTAIN	Kade2015	X	X			
FAR	Rothbaum2014			X		
INDUS	Mujber2004		X		X	X
JUE	Zyda2005	X	X	X		
MARK	Liu2007					X
MARK	Tussyadiah2016	X				X
MED	Baldominos2015	X			X	
MED	Kim2005		X	X	X	X
MED	Parsons2009	X		X	X	
MED	Popescu2000			X	X	
MED	Rizzo2001			X	X	
MED	Satava1998		X			X
MIL	NATO2003		X	X	X	X
MIL	Armstrong2013		X	X		X
MIL	Lele2013		X	X	X	
MIN	VanWyk2009		X	X		
NEUPSI	Rizzo1997	X		X	X	
NEUPSI	Rizzo2004	X		X	X	
NEURO	Tarr2002		X			X
REST	Hwang2012					X
RETA	Flanagan2014					X
RETA	Forbes2016					X
RETA	Guidi2010		X			X
ROBO	Burdea1999			X	X	X
TRANS	Erath2016				X	X
TRANS	Li2015	X			X	X
TRANS	Yu2013				X	X
TRANS	Yu2014					X
TURI	Guttentag2010		X			X
TURI	Huang2016		X			X
URBA	Nguyen2016				X	X

Figura 2.8: Matriz clusterizada

Por último vamos a denominar a cada grupo con un sustantivo que intente definir de la manera más general posible los casos de uso que engloba. Dichos nombres son arbitrarios y podrían ser objeto de debate pero independientemente del sustantivo utilizado el agrupamiento es estadísticamente demostrable.

### 2.2.9.2 Las cinco categorías de casos de uso de la Realidad Virtual

En base a los resultados hallados en los puntos anteriores se propone la siguiente taxonomía:



Figura 2.9: Las cinco categorías de los casos de uso

Visualización	<ul style="list-style-type: none"> <li>● evaluación prototipos</li> <li>● mejora comprensión/interpretación resultados</li> <li>● toma decisión</li> <li>● visualización big data</li> <li>● visualización datos</li> <li>● visualización datos grupal/data sharing</li> </ul>
Aprendizaje	<ul style="list-style-type: none"> <li>● ensayo acciones físicas/mentales</li> <li>● entrenamiento/training</li> <li>● experimentación ecológica</li> <li>● formación/learning</li> </ul>
Medicina	<ul style="list-style-type: none"> <li>● diagnóstico</li> <li>● mejora sensorial</li> <li>● tratamiento</li> <li>● rehabilitación/telerehabilitación</li> </ul>
Ocio	<ul style="list-style-type: none"> <li>● agentes humanos</li> <li>● lúdico</li> <li>● mejora inmersión</li> <li>● turismo virtual</li> <li>● virtual shopping</li> </ul>
Tecnología	<ul style="list-style-type: none"> <li>● diseño de materiales</li> <li>● simulación</li> <li>● sinergia tecnológica</li> <li>● teleoperación</li> </ul>

Tabla 2.7: Jerarquía de casos de uso

Los casos de uso quedarían organizados siguiendo la jerarquía descrita en la tabla superior.



En este momento podemos realizar una representación gráfica de la relevancia de cada categoría de uso en función del número total de referencias de las publicaciones estudiadas. De un total de 112 referencias totales, la distribución es la siguiente:

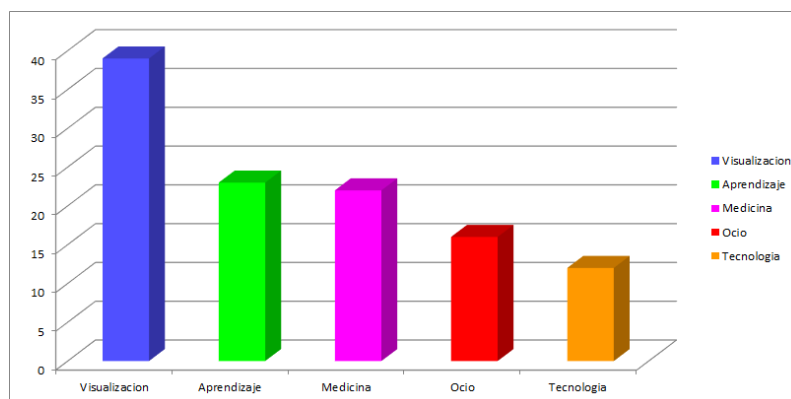


Figura 2.10: Distribución de casos de uso por categorías

Como se puede observar la Visualización es mayoritaria con un 35 % de las referencias. Después Aprendizaje y Medicina y por último Ocio y Tecnología.

Sin embargo no debemos de obviar el hecho de que, tal y como puede observarse en las matrices anteriores, los casos de uso de una publicación suelen estar dispersos entre 2 o incluso 3 áreas. Esto puede significar que no existe una separación estricta entre estas categorías de uso, o visto de otra manera, que tienen algún tipo de relación entre ellas.

### 2.2.9.3 Relaciones y dependencias entre grupos de uso de la Realidad Virtual

Basándonos en la matriz clusterizada anterior vamos a asignar cada publicación con dos nuevos valores: grupo primario y grupo secundario. Siendo el grupo primario el grupo de uso con mayor número de casos de uso de dicha publicación, y por lógica siendo el grupo secundario el segundo mayor número de casos de uso.

En los casos de «empate» se ha utilizado el contexto concreto de la publicación.

Esta es la tabla resultante:

Sector	Autor	Grupo1	Grupo2	Medicina	Aprendizaje	Visualizacion	Ocio	Tecnologia
AUTO	Berg2016	Visualizacion	Aprendizaje	0	1	2	0	1
AUTO	Stork2012	Aprendizaje	Visualizacion	0	2	1	0	0
EDU	Kaufmann2000	Visualizacion	Aprendizaje	0	1	2	0	0
EDU	Rizzo2006	Medicina	Aprendizaje	3	2	0	0	0
ENTERTAIN	Kade2015	Ocio	Aprendizaje	0	1	0	3	0
FAR	Rothbaum2014	Medicina	Aprendizaje	1	0	0	0	0
INDUS	Mujber2004	Visualizacion	Aprendizaje	0	2	3	0	1
JUE	Zyda2005	Aprendizaje	Ocio	0	3	0	1	0
MARK	Liu2007	Visualizacion	Ocio	0	0	1	1	0
MARK	Tussyadiah2016	Visualizacion	Ocio	0	0	1	1	0
MED	Baldominos2015	Medicina	Ocio	1	0	0	1	0
MED	Kim2005	Medicina	Visualizacion	2	2	2	0	0
MED	Parsons2009	Medicina	Ocio	3	0	0	1	0
MED	Popescu2000	Medicina	Aprendizaje	1	0	0	0	0
MED	Rizzo2001	Medicina	Aprendizaje	2	0	0	0	0
MED	Satava1998	Visualizacion	Aprendizaje	0	1	2	0	0
MIL	NATO2003	Tecnologia	Aprendizaje	0	3	2	0	3
MIL	Armstrong2013	Medicina	Aprendizaje	2	0	0	0	0
MIL	Lele2013	Aprendizaje	Visualizacion	0	2	1	0	1
MIN	VanWyk2009	Aprendizaje	Visualizacion	0	2	0	0	0
NEUPSI	Rizzo1997	Medicina	Ocio	3	0	0	1	0
NEUPSI	Rizzo2004	Medicina	Ocio	3	0	0	1	0
NEURO	Tarr2002	Aprendizaje	Visualizacion	0	1	1	0	0
REST	Hwang2012	Visualizacion	Ocio	0	0	2	0	0
RETA	Flanagan2014	Visualizacion	Tecnologia	0	0	2	0	0
RETA	Forbes2016	Visualizacion	Tecnologia	0	0	2	0	0
RETA	Guidi2010	Visualizacion	Ocio	0	0	2	1	0
ROBO	Burdea1999	Tecnologia	Medicina	1	0	0	0	3
TRANS	Erath2016	Visualizacion	Tecnologia	0	0	2	0	1
TRANS	Li2015	Visualizacion	Ocio	0	0	3	2	1
TRANS	Yu2013	Visualizacion	Tecnologia	0	0	2	0	0
TRANS	Yu2014	Visualizacion	Tecnologia	0	0	2	0	0
TURI	Guttentag2010	Ocio	Visualizacion	0	0	1	1	0
TURI	Huang2016	Ocio	Visualizacion	0	0	1	2	0
URBA	Nguyen2016	Visualizacion	Tecnologia	0	0	2	0	1

Figura 2.11: Matriz clusterizada por grupo primario y secundario

Primero vamos a estudiar la relaciones entre grupos de uso observando las combinaciones de «grupo primario» y «grupo secundario» que se producen. En la siguiente tabla se contabilizan las ocasiones en las que el grupo primario (primera columna) está seguido del grupo secundario (primera fila). De tal manera que la primera fila podría leerse: «El grupo medicina suele ir mayormente asociado al aprendizaje y en menor medida por el ocio».

Columna1 ▼	Medici▼	Aprendiza▼	Visualizaci▼	Oc▼	Tecnolog▼
Medicina	0	5	1	4	0
Aprendizaje	0	0	4	1	0
Visualización	0	4	0	5	6
Ocio	0	1	2	0	0
Tecnología	1	1	0	0	0

Figura 2.12: Tabla de relaciones entre grupos

De la observación de esta tabla podemos interpretar las siguientes aseveraciones:

- El grupo *medicina* suele ir mayormente asociado al *aprendizaje* y en menor medida por el *ocio*
- El grupo de *aprendizaje* casi siempre va asociado a la *visualización*
- La *visualización* está asociada fuertemente al resto de grupos
- El *ocio* está asociado mayormente a la *visualización* y en menor medida por el *aprendizaje*
- La *tecnología* está ligeramente asociada a la *medicina* y al *aprendizaje*

### 2.2.9.4 Casos de uso VR por Sector Económico

Volviendo al objetivo inicial del presente estudio, ya estamos en condiciones de estudiar cómo se distribuyen los casos de uso de **V.R** por sector. En primer lugar agrupamos los casos de uso por sector con el resultado de la siguiente tabla:

Sector	agentes humanos	lúdico	mejora inmersión	turismo virtual	virtual shopping	ensayo acciones físicas/mentales	entrenamiento/training	experimentación ecológica	formación/learning	diagnóstico	mejora sensorial	tratamiento	rehabilitación/tele-rehabilitación	diseño de materiales	simulación	sinergia tecnológica	teleoperación	evaluación prototipos	mejora comprensión/interpretación	toma decision	visualización big data	visualización datos	visualización datos grupal/data sharing
AUTO	0	0	0	0	0	1	2	0	0	0	0	0	0	1	0	0	0	2	0	1	0	0	0
EDU	0	0	0	0	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	0	0	0	1
ENTERTAIN	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FAR	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
INDUS	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	1
JUE	0	1	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MARK	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
MED	0	2	0	0	0	1	1	0	1	3	0	1	5	0	0	0	0	1	1	0	0	1	1
MIL	0	0	0	0	0	2	2	0	1	1	0	0	1	0	2	1	1	0	0	2	0	1	0
MIN	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NEUPSI	0	2	0	0	0	0	0	0	0	2	0	2	2	0	0	0	0	0	0	0	0	0	0
NEURO	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
REST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
RETA	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	3	0	3	0	0	0
ROBO	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0	0	0
TRANS	1	1	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	1	2	2	2	0	2
TURI	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
URBA	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0

Figura 2.13: Casos de uso por Sector Económico

Para hacer más comprensible esta información la trasladamos a un mapa de calor:

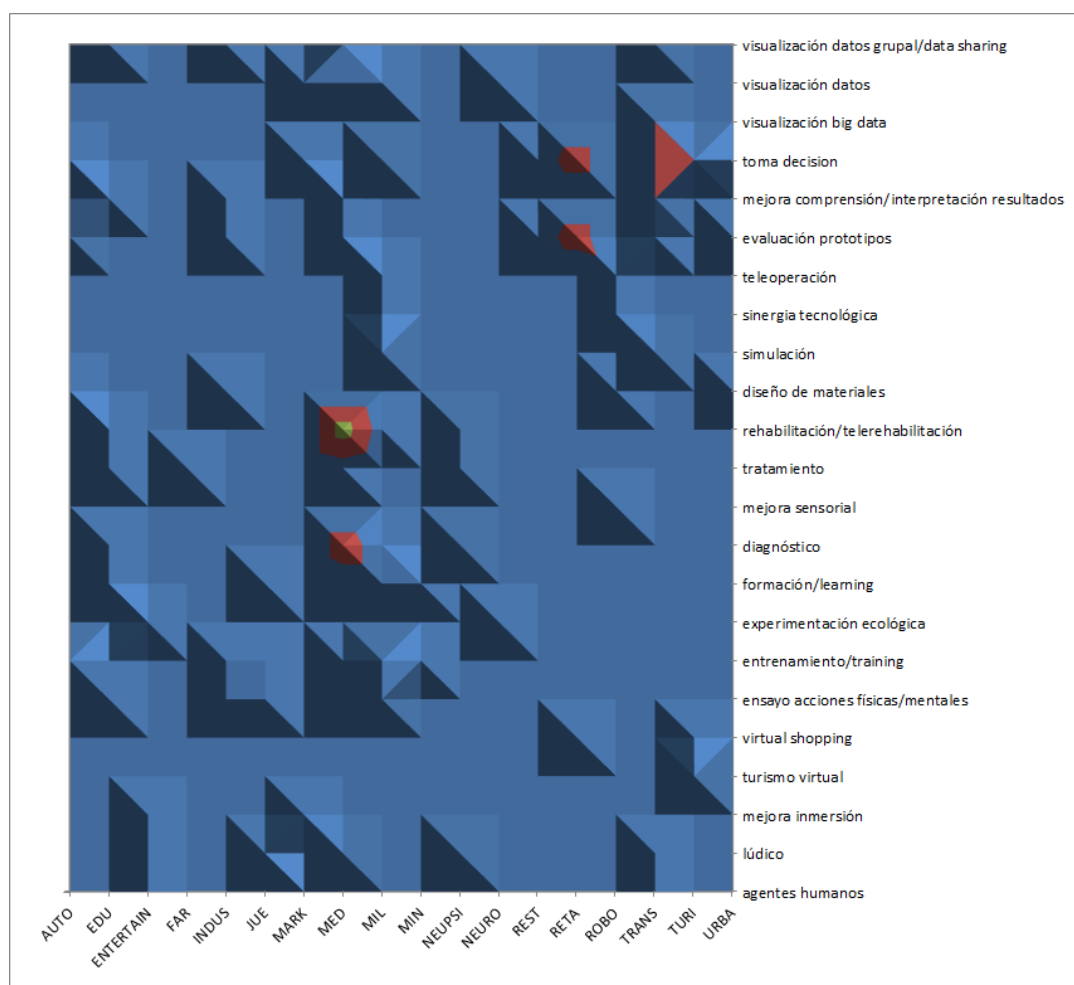


Figura 2.14: Mapa de Calor casos de uso por sector económico

En esta representación gráfica podemos observar varias «zonas calientes» en las intersecciones entre:

- **Medicina** y *rehabilitación, tratamiento y diagnóstico*
- **Retail** y *toma de decisión y evaluación de prototipos*
- **Transporte** y *mejora comprensión, toma decisión y visualización big data*

Podemos simplificar el gráfico anterior agrupando los resultados por grupo de caso de uso en lugar de verificar cada caso de uso por separado. El resultado sería la siguiente tabla:

Sector	AUTO	EDU	ENTERTAIN	FAR	INDUS	JUE	MARK	MED	MIL	MIN	NEUPSI	NEURO	REST	RETA	ROBO	TRANS	TURI	URBA
Visualización	3	2	0	0	3	0	2	4	3	0	0	1	2	6	0	9	2	2
Tecnología	1	0	0	0	1	0	0	0	4	0	0	0	0	0	3	2	0	1
Medicina	0	3	0	1	0	0	0	9	2	0	6	0	0	0	1	0	0	0
Aprendizaje	3	3	1	0	2	3	0	3	5	2	0	1	0	0	0	0	0	0
Ocio	0	0	3	0	0	1	2	2	0	0	2	0	0	1	0	2	3	0

Figura 2.15: Grupos de Casos de uso por Sector Económico

Y su correspondiente representación gráfica:

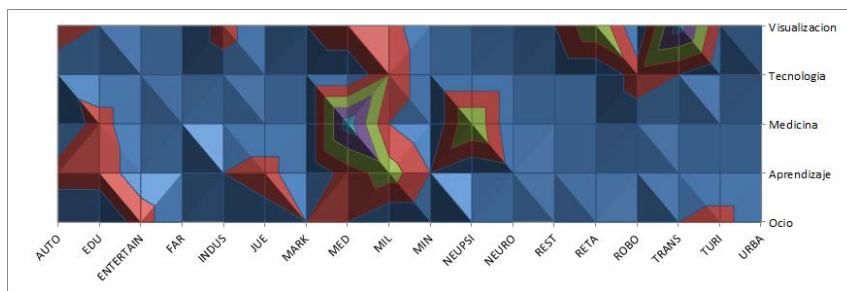


Figura 2.16: Mapa de Calor Grupo de Casos de uso por Sector Económico

Donde se vuelven a observar los patrones anteriores con un poco más de claridad. En general se observa una fuerte interrelación entre los casos de uso médico y el área de la sanidad (lo que parece lógico) y la neurociología. También el retail y el transporte y la visualización de datos.

Si intentamos representar las cinco categorías de casos de uso identificadas con un diagrama de Benn para representar sus interacciones potenciales podemos utilizar esta *rosa de los vientos*:



Figura 2.17: Interacciones de las cinco categorías de casos de uso

### 2.2.9.5 Ranking de sectores por casos de uso de la Realidad Virtual

Si revisamos el número de casos de uso totales por cada sector de actividad tendremos la siguiente gráfica:

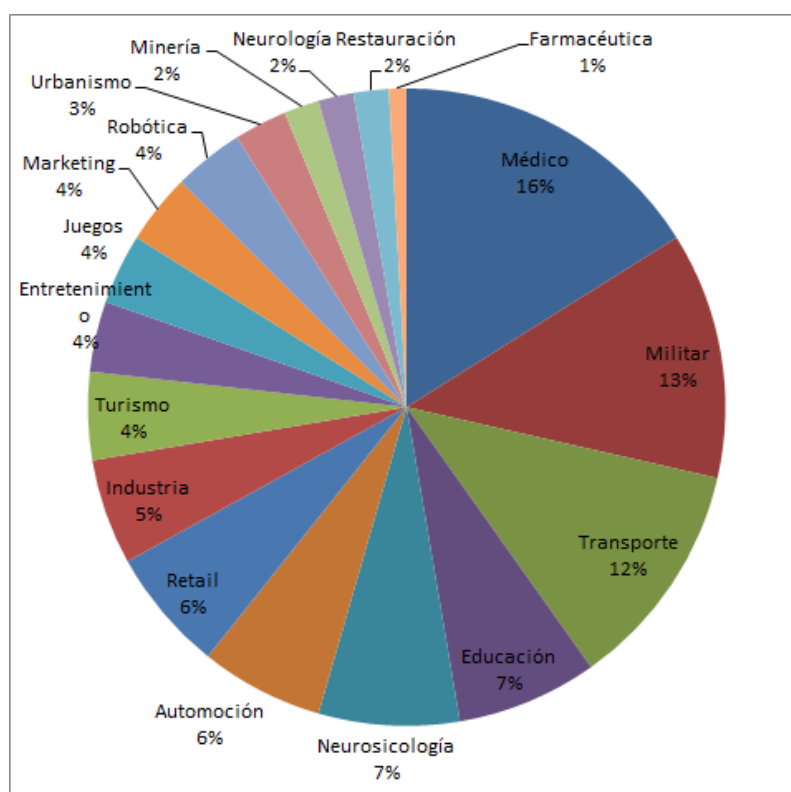


Figura 2.18: Ranking por sectores de casos de uso RV

Al aplicar el principio de Pareto, se observa que los sectores con mayor número de casos de uso **V.R** son el médico, el militar, el transporte, la educación y la neurociología.

Como ya se ha comentado anteriormente estas métricas hablan únicamente de publicaciones realizadas hasta la fecha y de casos de uso. No estamos midiendo volumen de negocio ni potencial de éxito ya que, algunos sectores, como por ejemplo la industria del Juego, tendrán seguramente un papel más relevante de lo que aparece.

Es sin embargo muy notable que los tres sectores que con diferencia tienen mayor número de aplicaciones sean estratégicos en las economías mundiales. Los sectores médico, militar y transporte son tres puntos clave para el control y la gestión de la economía. Las consecuencias derivadas de este hecho se escapan al alcance de esta tesis ya que no parece un problema baladí y requerirían de un estudio mucho más detallado.

### 2.2.9.6 Evolución sectores y casos de uso de la Realidad Virtual

Tras haber estudiado los casos de uso, sus grupos y cómo se relacionan con los sectores de actividad más importante es el momento de añadir la dimensión temporal a la observación. Para ello vamos a agrupar el número de casos de uso identificados en cada publicación por año. El rango temporal que vamos a usar es el que va desde 1997 (año de la primera publicación estudiada) hasta 2016, es decir 19 años. Existen dos años en esta secuencia (2008 y 2011) en los que no se recogen publicaciones aunque este hecho no nos impide comprobar las tendencias.

La tabla de casos de uso **V.R** a lo largo de los años es la siguiente:

Año	agentes humanos	lúdico	mejora inmersión	turismo virtual	virtual shopping	ensayo acciones físicas/mentales	entrenamiento/training	experimentación ecológica	formación/learning	diagnóstico	mejora sensorial	tratamiento	rehabilitación/tele-rehabilitación	diseño de materiales	simulación	sinergia tecnológica	teleoperación	evaluación prototipos	mejora comprensión/interpretación	toma decision	visualización big data	visualización datos	sharing
1997	0	1	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0
1998	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
1999	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0	0	0
2000	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	1
2001	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
2002	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
2003	0	0	0	0	0	1	1	0	1	0	0	0	0	0	1	1	1	0	0	1	0	1	0
2004	0	1	0	0	0	1	1	0	0	1	0	1	1	1	0	0	0	1	1	0	0	0	1
2005	0	1	0	0	0	2	1	0	2	1	0	0	1	0	0	0	0	1	1	0	0	0	0
2006	0	0	0	0	0	1	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0
2007	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
2008	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2009	0	1	0	0	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0
2010	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0	0	0
2011	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2012	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	2	0	1	0	0	0
2013	0	0	0	0	0	1	1	0	0	1	0	0	1	0	1	0	0	0	1	1	0	1	1
2014	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	1	0	1
2015	2	3	1	0	0	0	0	1	0	0	0	0	1	0	1	0	0	1	1	1	0	0	0
2016	1	0	0	1	1	0	1	0	0	0	0	0	0	2	1	0	0	3	1	6	0	0	0

Figura 2.19: Casos de Uso en el tiempo



Debido al alto número de casos de uso representados el único gráfico que se muestra útil para mostrar la información sería un mapa de calor, pero en este caso tampoco es realmente ilustrativo con lo que es necesario simplificar el esquema y agruparemos por tipos de casos de uso tal y como hemos hecho anteriormente. La tabla resultante es la siguiente:

Visualización	0	2	0	2	0	1	2	3	2	0	1	0	0	3	0	3	3	4	3	10
Tecnología	0	0	3	0	0	0	3	1	0	0	0	0	0	0	0	0	1	0	1	3
Medicina	3	0	1	1	2	0	0	3	2	3	0	0	3	0	0	0	2	1	1	0
Aprendizaje	0	1	0	1	0	1	3	2	5	2	0	0	2	0	0	2	2	0	1	1
Ocio	1	0	0	0	0	0	0	1	1	0	1	0	1	2	0	0	0	0	6	3
Año	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016

Figura 2.20: Grupos de Casos de Uso en el tiempo

Y su representación gráfica:

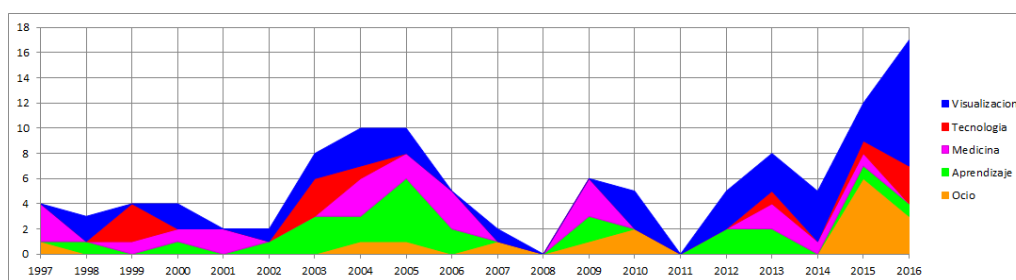


Figura 2.21: Gráfica Grupos de Casos de Uso en el tiempo

Como puede observarse rápidamente el grupo de casos de uso relativo a la «Visualización» es, con diferencia, el que más está creciendo de manera sostenida desde los últimos dos años. También el «Ocio» y la «Tecnología» están teniendo bastante auge.

Por último debemos constatar que la misma tendencia que se detectaba en la primera parte de este estudio (desde 2014 se ha disparado el interés en la **V.R**) se refleja en esta gráfica. Sólo en 2016 hay 17 referencias a casos de uso mientras que de media en años anteriores rondan las 5 referencias.

Sintetizando al máximo las conclusiones podemos enumerar las siguientes aseveraciones:

- Los sectores económicos donde la **V.R** es más aplicable son: **Medicina, Militar y Transporte**
- La aplicación de la **V.R** con mayor crecimiento y potencial es: **Visualización**

### 2.2.10 Problemas Identificados

Si bien es cierto que en la mayor parte de las publicaciones que se han revisado el tono general es de aceptación y muchas expectativas respecto de esta tecnología, no podemos obviar que también existen problemas serios que ponen en peligro el futuro de la misma.

Para identificar correctamente estos problemas se ha realizado un estudio similar al que hemos llevado a cabo en 2.2.9 revisando todos y cada uno de los problemas identificados en las publicaciones a estudio, ordenándolos y categorizándolos para su mejor comprensión.

En las siguientes secciones se documenta el estudio realizado, las estadísticas extraídas y finalmente se plantean y describen los problemas existentes, ordenados por importancia.

#### 2.2.10.1 Lista de problemas identificados

En concreto se han encontrado 35 problemas a lo largo de 15 publicaciones en las que se describen. Son los siguientes:

- cables y pantallas
- efectos secundarios
- hándicap sensorial (olfato, gusto)
- inmersión insuficiente
- retroalimentación háptica insuficiente
- mayor interacción con mundos virtuales
- métodos de interacción inmaduros
- mejoras en la modelación 3d
- rendeo de modelos más grandes
- dificultad construir EV física y geométricamente exactos
- falta de metáforas antropomórficas
- aumentar la validez ecológica
- aumentar calidad inmersiva
- falta metodología de diseño
- falta metodología explotación
- falta reconocimiento académico como disciplina
- proceso de ingeniería inmaduro
- validez ecológica cuestionable
- falta de una plataforma genérica, accesible y unificada
- falta de compatibilidad de la plataformas
- falta facilidad acceso para público no experto

## 2.2.10.2 Matriz de Publicaciones / Problemas

En la presente matriz se muestran las relaciones entre las publicaciones y los problemas identificados en cada una de ellas. Dado que no en todas las publicaciones se identifican claramente los problemas encontrados, al contrario que con los casos de uso, representaremos en el eje horizontal las publicaciones y en el eje vertical los problemas encontrados, ya categorizados por su campo de aplicación.

En este sentido se han identificado cuatro campos de aplicación: **hardware**, **software**, **metodología** y **servicios**. En la intersección entre publicaciones y problemas se coloca una «X» si en dicha publicación se ha encontrado el problema indicado.

		Sector	AUTO	AUTO	EDU	EDU	ENTE	FAR	INDU	JUE	MARK	MARK	MED	MED	MED	MED	TURI
		Autor	Berg2016	Stork2012	Kaufmann2000	Rizzo2006	Kade2015	Rothbaum2014	Mujber2004	Zyda2005	Liu2007	Tussyadiah2016	Baldominos2015	Kim2005	Parsons2009	Rizzo2001	Guttentag2010
Hardware	cables y pantallas													X			
	efectos secundarios		X		X								X	X			
	handicap sensorial (olfato, gusto)								X		X						X
	inmersión insuficiente									X		X	X		X		
	retroalimentación háptica insuficiente		X													X	
Software	mayor interacción con mundos virtuales		X					X									
	métodos de interacción inmaduros													X			
	mejoras en la modelación 3d		X														
	rendeo de modelos más grandes		X														
	dificultad construir EV física y geom. exactos			X													
	falta de metáforas antropomórficas											X					
	aumentar la validez ecológica														X		
	aumentar calidad inmersiva											X					
Metodología	falta metodología de diseño													X			
	falta metodología explotación													X			
	falta reconocimiento académico como disciplina									X							
	proceso de ingeniería inmaduro								X					X			
	validez ecológica cuestionable					X	X										
Servicios	falta de una plataforma genérica, accesible y unificada									X					X		
	falta de compatibilidad de la plataformas													X			
	falta facilidad acceso para público no experto								X					X			

Figura 2.22: Matriz expandida de publicaciones y los problemas detectados

Aunque el mayor número de problemas parece que se encuentran en la categoría "software", es cierto que algunos de dichos problemas son de similar naturaleza. Se ha realizado el ejercicio de condensar los problemas concretos en problemas generales de la siguiente forma:

*interacción inmadura* { *mayor interacción con mundos virtuales*  
*métodos de interacción inmaduros*  
*percepción inmadura* { *mejoras en la capacidad de modelado 3d*  
*rendeo de modelos más grandes*  
*dificultad construir entornos virtuales física y geométricamente exactos*  
*falta de metodologías* { *falta metodología de diseño*  
*falta metodología explotación*

Quedando la misma matriz simplificada de la siguiente forma:

		Sector	AUTO	AUTO	EDU	EDU	ENTE	FAR	INDU	JUE	MARK	MARK	MED	MED	MED	MED	TURI
		Autor	Berg2016	Stork2012	Kaufmann2000	Rizzo2006	Kade2015	Rothbaum2014	Mujber2004	Zyda2005	Liu2007	Tussyadiah2016	Baldominos2015	Kim2005	Parsons2009	Rizzo2001	Guttentag2010
Hw	efectos secundarios		1	1									1	1			
	handicap sensorial (olfato, gusto)								1		1			1			1
	inmersión insuficiente									1		1	1		1		
	retroalimentación háptica insuficiente		1													1	
Sw	interacción inmadura		1					1						1			
	percepción inmadura		2	1													
	falta de metáforas antropomórficas											2			1		
Met	falta metodología													2			
	proceso de ingeniería inmaduro								1	1				1			
	validez ecológica cuestionable					1	1										
Serv	falta de una plataforma genérica, accesible y unificada										1						
	falta de compatibilidad de la plataformas													1	1		
	falta facilidad acceso para público no experto								1					1			

Figura 2.23: Matriz reducida de publicaciones y los problemas detectados

Si agrupamos aún más por campo de aplicación tendremos la siguiente tabla:

	Sector	AUTO	AUTO	EDU	EDU	ENTE	FAR	INDU	JUE	MARK	MARK	MED	MED	MED	MED	TURI
	Autor	Berg2016	Stork2012	Kaufmann2000	Rizzo2006	Kade2015	Rothbaum2014	Mujber2004	Zyda2005	Liu2007	Tussyadiah2016	Baldominos2015	Kim2005	Parsons2009	Rizzo2001	Guttentag2010
Hardware		2		1			1	1	1	1	1	2	2	1	1	1
Software		3	1								2		1	1		
Metodología					1	1		1	1				3			
Servicios								1		1			2	1		

Figura 2.24: Matriz de publicaciones y tipos de problema

### 2.2.10.3 Análisis de las estadísticas de problemas detectados

Si acumulamos los problemas detectados por campo de aplicación y los representamos en una gráfica obtendremos la siguiente distribución:

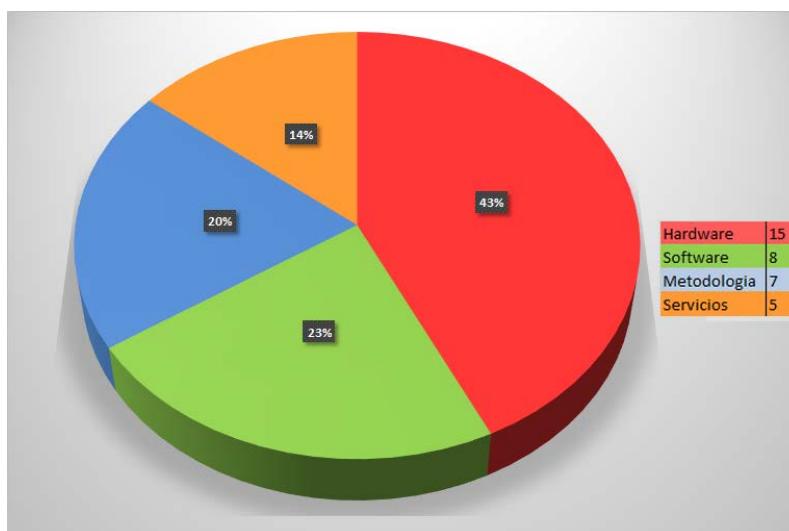


Figura 2.25: Distribución porcentual por campo de aplicación

Como puede observarse, los problemas asociados al **hardware** son mayoritarios, seguidos por los problemas relativos al **software**, a las **metodologías** y por último los **servicios**.

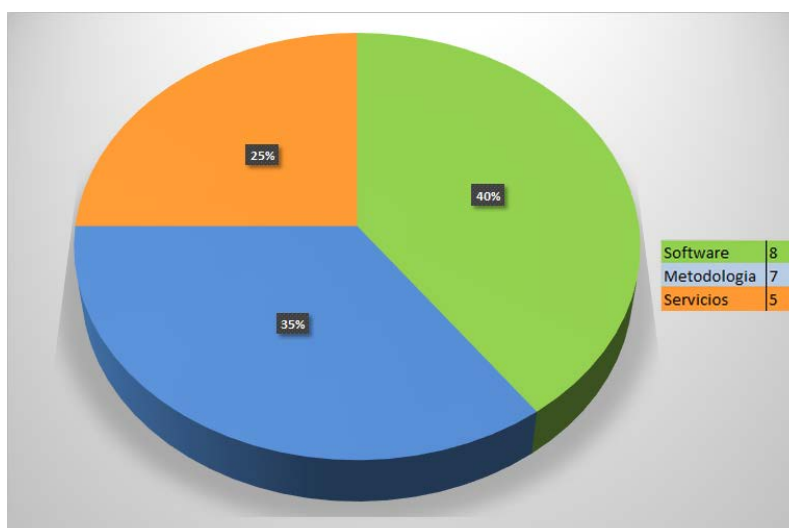


Figura 2.26: Distribución porcentual por campo de aplicación excepto hardware

Es completamente lógico que los problemas **hardware** sean los más abundantes

cuando se está en una fase inmadura del desarrollo de una tecnología, sin embargo pueden dar una imagen sesgada de la realidad de los problemas existentes. Si eliminamos la categoría de problemas **hardware** obtendremos la siguiente distribución.

Otro factor más que debemos tener en cuenta es el año de publicación de los artículos. En los últimos 17 años la tecnología **hardware** de la **V.R** ha evolucionado de manera drástica, así que vamos a estudiar la evolución de los problemas a lo largo del tiempo mediante esta gráfica:

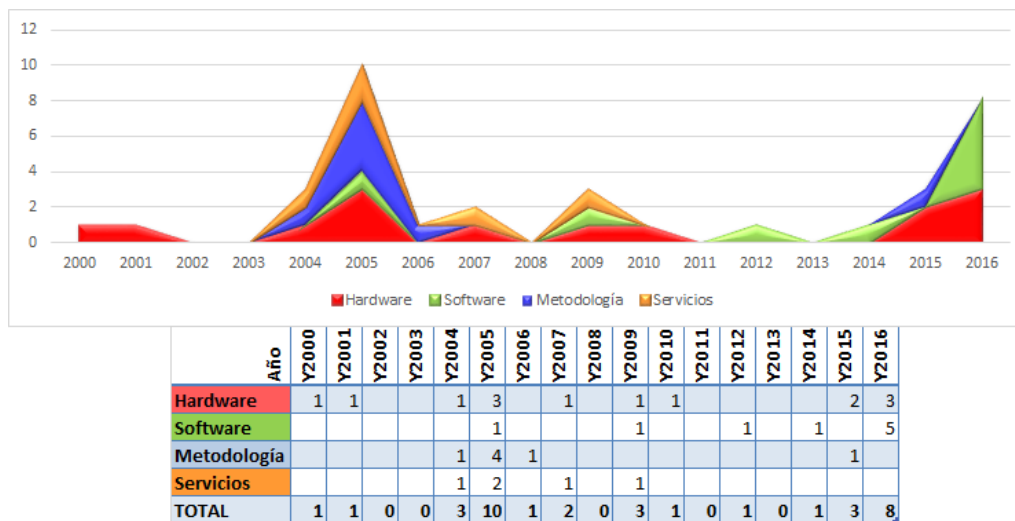


Figura 2.27: Evolución de los problemas en el tiempo

De esta representación de datos podemos desprender que los errores **hardware** eran los más abundantes al comienzo. Sin embargo en los últimos años, los problemas más detectados por los investigadores se refieren en primer lugar al **software** y en segundo lugar a la **metodología**.

#### 2.2.10.4 Resumen de los problemas no-hardware más importantes

Dado el contexto de la presente tesis vamos a dejar de lado los problemas que están derivados única y exclusivamente del **hardware**. Estos suelen estar derivados de la falta de capacidad de cálculo, miniaturización y accesibilidad de los equipos **V.R.** El estudio e investigación en este campo es muy extenso y no entra del alcance de la presente tesis.

Nos vamos a centrar en los otros problemas relacionados con el **software**, la **metodología** y los **servicios**:

Software	<ul style="list-style-type: none"> <li><i>interacción inmadura</i></li> <li><i>percepción inmadura</i></li> <li><i>falta de metáforas antropomórficas</i></li> </ul>
Metodología	<ul style="list-style-type: none"> <li><i>proceso de ingeniería inmaduro</i></li> <li><i>falta metodología</i></li> <li><i>validez ecológica cuestionable</i></li> </ul>
Servicios	<ul style="list-style-type: none"> <li><i>falta de compatibilidad de la plataformas</i></li> <li><i>falta facilidad acceso para público no experto</i></li> <li><i>falta de una plataforma genérica, accesible y unificada</i></li> </ul>

Si los ordenamos por la frecuencia en la que aparecen en las publicaciones estudiadas obtendremos la siguiente gráfica:

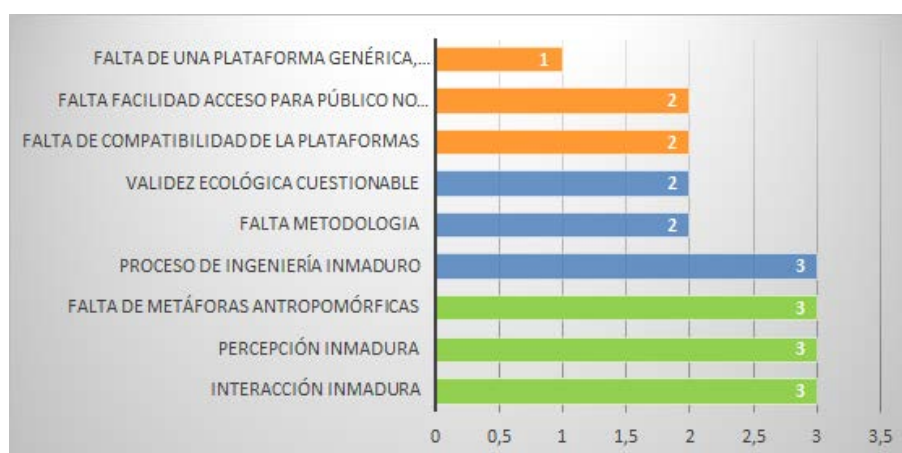


Figura 2.28: Ranking de problemas

## 2.3 Sistemas Multiagente

### 2.3.1 Introducción

Tal y como se amplía en los puntos 1.4.9 y 3.12 una de las propuestas más importantes que se realiza en esta tesis es la de utilizar el paradigma de programación orientada a agentes, el modelo multiagente y en particular los sistemas multiagente situados como la herramienta idónea para abordar la construcción de **V.I.E's**.

Es por tanto imprescindible en este momento realizar un repaso a la situación actual y comprender qué iniciativas existen hoy en día en este campo.

Cuando hablamos de sistemas multiagente siempre nos expresamos en términos muy humanizados: decimos que ciertos agentes se desenvuelven en... o que los agentes hablan entre ellos para... Esta es una actitud completamente razonable ya que la naturaleza propia de los agentes, su independencia, robustez, comportamiento social, etc. es compartida por los seres humanos.

Solemos dar por supuesto que dichos agentes se desenvuelven en un entorno que habitualmente no es nuestro contexto. Por eso cuando decimos «el agente 1 le manda un mensaje al 2» sin dar información de contexto estamos cometiendo un error. ¿Cómo se manda? ¿Qué mecanismo se usa? Es significativamente interesante que esta idea ha estado desde el primer momento en toda la investigación al respecto de los MAS, pero siempre en un segundo plano. Se ha tratado al entorno como un simple sustrato sin importancia.

Al menos ha sido así en la mayor parte de los casos ya que desde hace unos años una serie de investigadores se han dado cuenta de este problema y han realizado un importante avance en los MAS al incluir el entorno como parte fundamental del sistema.

En este capítulo nos centraremos en la razones, ventajas y modelos que hacen del entorno un elemento de primer orden en los MAS situados, para posteriormente presentar distintos casos prácticos tanto de MAS situados como de entornos virtuales simulados inmersivos. Este capítulo se cierra con un resumen en el que se sintetizan las conclusiones y las posibilidades futuras para esta fusión tecnológica y conceptual.



### 2.3.2 Inicios

En 1990 Steels introduce (Steels, 1990) los «campos de gradiente» del entorno como mecanismo básico para que los agentes puedan comunicarse y coordinarse a través del mismo.

En 1994, Maes (Maes and Maes, 1994) asocia los agentes reactivos a los principios de la robótica en su aspecto dependiente del entorno. En este periodo de tiempo varios autores como Drogoul (Drogoul and Ferber, 1992), Dorigo (Dorigo and Gambardella, 1997) y Parunak (Parunak, 1997) se inspiran en las colonias de insectos (hormigas, abejas, etc.) para extraer modelos computacionales que establezcan las bases sociales de los agentes y su entorno.

Una de las primeras publicaciones en las que se menciona explícitamente la importancia del entorno en los Sistemas Multiagente es la de Wooldridge y Jennings (Wooldridge and Jennings, 1995) en 1995 donde se define agente como: «*un sistema computacional que está emplazado en un entorno, y que es capaz de realizar acciones autónomas en su entorno con el objetivo de conseguir los objetivos con los que fue diseñado*». En esta definición, la situación expresa el importante hecho que un agente no es una entidad aislada sino que existe dentro de un entorno, aunque no se define claramente lo que significa entorno. Sucede por lo tanto algo habitual cuando se define un elemento en función de otro que no está bien definido: que cada cual entiende el concepto libremente. En este caso, el término «entorno» no es un concepto desconocido, más al contrario tiene muchas distintas acepciones y esa es la razón principal de la falta de acuerdo.

### 2.3.3 Ferber y Muller

En (Ferber and Müller, 1996) Ferber y Muller marcan el inicio de los MAS situados tal y como los estamos estudiando hoy en día. En dicho trabajo se define un modelo específico para los **S.M.A.S** enunciando que existen dos tipos de interacción entre los agentes situados: las comunicaciones entre un agente y su entorno y aquellas comunicaciones que se realizan entre agentes a través del entorno.

Como base a este modelo utilizan una variante del «situation calculus». En este formalismo lógico se asume que el mundo está en un estado determinado que sólo cambia como resultado de un agente que realiza una acción, a su vez el estado está definido a

través de los predicados y los argumentos de las funciones. En otras palabras, aceptamos la asunción de (Reiter, 1978, 1991) mundo cerrado.

Aun así es necesario modificar el modelo base dividiendo las acciones en dos partes: las precondiciones y los efectos. El axioma precondición define las condiciones que han de ser obligatoriamente cumplidas para que la acción pueda ser tomada por el agente, y en el axioma de los efectos se describe las modificaciones al estado del entorno que la ejecución de dicha acción produce.

Una de las observaciones más interesantes son las «leyes» del entorno. Dichas leyes son uno de los primeros ejemplos de **S.M.A.S** en las que las modificaciones al entorno no vienen producidas por los agentes sino por el propio entorno, que tiene ciertas normas de obligado cumplimiento para todos los agentes. Un ejemplo, de estas leyes podría ser expresado de la siguiente manera (Ferber and Müller, 1996):

```
<displace(x,y),
{hold(x,y), height(y,h)},
{apply(x,up)},
{-height(y,h), height(y,h+(up+weight))}>
<tilt(y),
{height(left(y),h1), height(right(y),h2), tilt(y,old))},
{},
{-tilt(y,old),tilt(y,arcsin((h1h2)/d(left(y),right(y)))}>
```

Finalmente se distingue entre dos tipos de agentes: tropísticos e histeréticos, que no es sino otra manera de denominar respectivamente reactivos y deliberativos.

### 2.3.4 Bandini, Manzoni y Simone

Otro interesante modelo de MAS Situado ha sido propuesto por Bandini, Manzoni y Simone en (Bandini et al., 2002). La investigación de las científicas italianas se centra en los MAS situados que específicamente usan un entorno con estructura espacial. Tal y como se argumenta, este es el primer trabajo que hace un especial hincapié en la componente espacial de los entornos propios a los **S.M.A.S**.

También definen un nuevo concepto compuesto: el sistema multiagente situado multicapa, o Mmass según sus siglas en inglés (Multilayered Multi-Agent Situated

System). Estos Mmass están compuestos por un conjunto de MASS simples que cumplen las siguientes premisas:

- Contiene agentes situados.
- Las interacciones de los agentes con los Mmass son dependientes del espacio, o lo que es lo mismo, se desarrolla a través del espacio propio del entorno.
- El estado y la posición del agente pueden ser modificados por el agente de acuerdo a un mecanismo de percepción-deliberación-acción.
- Los agentes son, o al menos pueden ser, heterogéneos como resultado de la naturaleza múltiple de los Mmass.

Además de estas premisas se construye una definición del entorno según estas reglas:

- Un MAS situado y multicapa se compone por la tripla:  $\langle Space, F, A \rangle$
- Donde a su vez  $Space$  se define como:  $\langle \langle Space_1, F_1, A_1 \rangle, \langle Space_2, F_2, A_2 \rangle, \dots, \langle Space_n, F_n, A_n \rangle \rangle$
- Donde a su vez:  $Space_x$  es un espacio del total del espacio de espacios,  $F_x$  es un Field (campo) de entre el total del espacio de campos y  $A_x$  es un agente de entre los agentes existentes en dicho MASS.

Por lo tanto, un Mmass es un conjunto de los espacios, los campos y los agentes.

Para definir  $Space$  (espacio) es necesario dos cosas: la primera es definir  $P$  como lugar (place) que designa un punto espacial concreto del entorno. La segunda es redefinir el espacio de nuevo como la tripla:  $\langle A_p, F_p, P_p \rangle$ , donde:

- $A_p$  es el agente situado en  $p$ . Puede haber cero o un agentes en determinado  $P$ .
- $F_p$  es conjunto de campos que están activos en  $P$ . Puede ser vacío.
- $P_p$  es el conjunto de sitios adyacentes a  $P$ .

En resumen, un espacio (Space) sería un grafo no orientado de sitios (P)

El concepto de Field (campo) es una abstracción para servir como contenedor de cualquier tipo de interacción que exista en el sistema. Cada campo está caracterizado por un conjunto de valores que definen su propagación por el espacio. Dichos valores son una 4-tupla:  $\langle W_f, Difussion_f, Compare_f, Compose_f \rangle$  donde:

- $W_f$  es el conjunto de valores que un campo puede tener.
- $Difussion_f : P \times W_f \times P \rightarrow (W_f)^+$  es decir, expresa la manera que los valores  $W_f$  van cambiando a lo largo del grafo de sitios P definido en el espacio.
- $Compare_f : W_f \times W_f \rightarrow \{True, False\}$  es decir, la función según se comparan dos campos dados para definir la percepción de los mismos.
- $Compose_f : (W_f)^+ \rightarrow W_f$  expresa la manera en la que los campos se combinan, por ejemplo para poder agregar todos los campos actuantes en un determinado sitio.

El conocido concepto de agente también tiene su propia abstracción. Se entiende que cada espacio de cada MASS puede albergar un conjunto de agentes definido por la tripla:  $\langle s, p, t \rangle$  dónde:

- $t$  es el tipo de agente
- $s$  es el estado del agente y puede tener un sólo valor de todos los valores definidos por su tipo
- $p$  es el sitio (place) del espacio (space) donde se encuentra

A su vez el tipo de agente sirve para describir los posibles estados y la sensibilidad del agente a los distintos campos, así como las acciones que se pueden realizar. El tipo de agente se define, como ya es habitual por otra tripla:  $\langle \Sigma_t, Perception_t, Action_t \rangle$  donde:

- $\Sigma_t$ : es el conjunto de estados que un agente del tipo  $t$  puede asumir

- $Perception_t$ : es la función que asocia a cada estado del agente un vector de pares tal que para el par  $i$ -ésimo, el primer elemento expresa un coeficiente que será aplicado al valor de campo y el segundo expresa la sensibilidad del agente a ese mismo valor de campo.
- $Action_t$ : define un conjunto de acciones que los agentes de tipo  $t$  pueden realizar

Por último definen las acciones (Actions) como el modelo que siguen los agentes para cambiar su estado y/o posición, cómo interactúan con otros agentes y como el entorno afecta a estos. Generalmente las acciones tienen dos posibles propósitos: modificar el estado o la posición de sí mismos o interaccionar con otros agentes (comunicación intra-agente o inter-agente).

Las acciones internas pueden ser a su vez de dos tipos «trigger» y «transport» mientras que las externas pueden ser «emit» y «reaction»:

- trigger: cambia el estado interno del agente
- transport: cambia la posición del agente
- emit: cambiar valor de difusión de un campo dado
- reaction: ante un determinado estado de los campos del entorno se desencadena un cambio de los otros tres tipos

### 2.3.5 Weyns et al.

Sin duda el modelo mejor definido y más trabajado es el de Weyns y otros en (Weyns, 2010; Weyns et al., 2009; Weyns and Holvoet, 2004; Weyns and Michel, 2015b; Weyns et al., 2007a, 2005, 2006, 2007b). En dicho estudio se sientan las bases de los **S.M.A.S** muy detalladamente, por ejemplo a la hora de definir cómo los elementos de un sistema real encajan en un modelo MAS: se entiende que cualquier elemento de un sistema multiagente (MAS) que no sean puramente agentes, son parte del entorno. Cualquier servicio web, infraestructura de comunicaciones o la topología del dominio espacial, es parte del entorno de un **S.M.A.S**.

El papel del entorno en los agentes cognitivos está relacionado con el modelo cognitivo que el agente tiene del mismo, las acciones que el agente puede realizar sobre el entorno

y el razonamiento práctico que tiene sobre dichas acciones. Desde el punto de vista de un agente individual es fácil pensar que el entorno es todo lo que está fuera del agente, incluidos los demás agentes. Por eso, los MAS basados en modelos cognitivos suelen tener dependencias sobre los objetivos para poder resolver los conflictos que se generan al tener muchos agentes compartiendo el mismo objetivo. Este es un problema que se repite siempre en estos sistemas y que obliga a tener en cuenta estos factores a la hora de diseñar las reglas de objetivo. Desde el punto de vista de los MAS situados (**S.M.A.S**) la coordinación de los objetivos queda bajo la responsabilidad del entorno.

Weyns et al. propone cinco perspectivas diferencias en cuanto al papel que el entorno tiene al trabajar con sistemas de múltiples agentes cognitivos que se describen a continuación:

#### **2.3.5.1 El entorno como contenedor y medio de comunicación**

Normalmente se piensa que el entorno es simplemente un lugar donde se almacenan los recursos de los agentes y un sistema de comunicación. Por contra, la pieza fundamental sobre la que se construye el modelo (FIPA, 2002) es el «agent platform» que consiste en:

- un directorio facilitador (yellow pages) para publicar y encontrar servicios
- un sistema de gestión de agentes (white pages) para permitir a los agentes registrarse en una plataforma o cambiar a otra
- un sistema de mensajería para poder mantener las comunicaciones entre los agentes y con los otros dos servicios

Por ejemplo la plataforma JADE (Bellifemine et al., 2001) que es una conocida herramienta de diseño de sistemas multiagente provee de estos tres servicios.

#### **2.3.5.2 El entorno como una capa organizacional**

La idea de tener un sistema multiagente, con sus muchos elementos, prácticamente implica la necesidad de disponer de un sistema de sea capaz de mantener un orden sobre estos. Este orden suele estar expresado como un conjunto de reglas y restricciones al comportamiento de los agentes. Si estas reglas se aplican en conjunto y sobre todo si se

localizan a una zona del entorno se puede hablar de «áreas». En Ferber (Ferber, 1999) se distingue entre dos tipos de MAS según su organización: centrados en el agente y centrados en la organización. En aquellos sistemas centrados en el agente este concepto de área se difumina, mientras que en los centrados en la organización se focaliza en el entorno. Aunque las comentadas reglas pueden estar embebidas en los agentes esta tarea es claramente una responsabilidad del entorno.

### 2.3.5.3 El entorno como infraestructura de coordinación para agentes cognitivos

Los sistemas de pizarra compartida fueron unas de las primeras aproximaciones para abordar el problema del control al construir sistemas distribuidos (Corchado and Molina, 2002), aunque posteriormente fueron evolucionando hacia otros modelos más complejos. Sin embargo la sencilla pizarra expresa perfectamente el potencial del entorno como herramienta de coordinación entre agentes. Además los entornos pueden tomar un papel coercivo, no solamente declarativo. Si la responsabilidad de mantener la coordinación se mantiene únicamente en los agentes no existe un control real. Pero en los **S.M.A.S** la responsabilidad de hacer cumplir las normas queda en el entorno, ya sean reglas físicas impuestas por la naturaleza (la gravedad en un entorno físico) o reglas específicas cognitivas como la organización eficiente de los objetivos de los agentes. Por ejemplo, si en una simulación **S.M.A.S** la velocidad máxima de los agentes es fundamental para evaluar su fitness es más lógico que el entorno penalice a aquellos agentes que sobrepasen el límite que los propios agentes tengan que marcar su tope de velocidad.

### 2.3.5.4 Entornos de Markov

Los modelos de decisión de Markov (MDP) son un modelo muy extendido para resolver problemas computacionalmente complejos. Básicamente un MDP consiste en un conjunto de estados, otro de acciones, otro de triplas estado-acción-estado' y finalmente otro conjunto de recompensas. El mayor desafío de este modelo es poder distinguir el espacio de estados existente en un problema dado. Esta información ha de ser extraída precisamente del entorno, por ejemplo en la competición de aprendizaje por refuerzo de 2009 (Reinforcement Learning 2009 competition) se planteó la posibilidad de que un agente utilizando MDP sea capaz de jugar a una versión libre del juego Super Mario. En (Mohan

and Laird, 2009) se explica cómo el entorno es todo el nivel (track) de juego y que el tamaño del espacio de estados era tan grande que tuvieron que reducirlo a unas pocas celdas adyacentes a Mario. Este es un claro ejemplo de problema en el que el espacio de estados es demasiado grande para ser tratado directamente tomando información sin tratar del entorno. Si una vez más traspasamos la responsabilidad de discretizar el espacio del problema al entorno obtendremos una arquitectura más clara. Existe un tipo diferenciado de MPD denominado «modelo de decisión de Markov parcialmente observable» (POMDP) en el que aún tiene más sentido la responsabilidad del entorno. Estos POMDP parten de que es imposible observar todo el espacio markoviano subyacente y por lo tanto es necesario realizar una abstracción estadística sobre el espacio de estados real. Esto se traduce en que en algunas implementaciones (Steels, 1990) es necesario realizar un estudio estadístico periódico para la declaración previa de estados observables, tarea que obviamente no debe ser desempeñada por un agente sino por el propio entorno.

#### 2.3.5.5 Entornos de tarea

Los entornos orientados a las tareas definen las características del entorno en el cual los agentes se desenvuelven y el conjunto de tareas que los agentes deben acometer en ese entorno. Entendemos tarea como un conjunto de objetivos que deben de cumplirse, o lo que es lo mismo, un agente debe intentar mantener ciertos estados en un determinado valor prefijado. Si esos estados cumplen ese valor, la tarea está cumplida. En este sentido el estado puede ser interno al agente o externo al mismo (Bandini et al., 2002), en este último caso el cumplimiento de la tarea está directamente influida por el valor de un estado intrínsecamente del entorno.

Como se ha expuesto en las cinco anteriores categorizaciones, se plantea el desplazamiento de las responsabilidades desde los agentes al propio entorno. Desde este punto de vista podemos pensar en el entorno como un «proveedor de soporte». Se proponen estos cuatro distintos niveles de soporte (esta idea se repite en otros autores (Bandini et al., 2001; Platon et al., 2007)):

- **Nivel básico.** En sustrato base, el entorno permite a los agentes acceder al nivel de despliegue. El nivel de despliegue comprende todos los elementos hardware y software de bajo nivel o elementos externos al sistema, por ejemplo cámaras, actuadores, impresoras, bases de datos, servicios web, etc.). Proveer de este tipo de



acceso es fundamental para los agentes y por lo tanto es el primer nivel de acceso que un entorno debe ofrecer.

- **Nivel de abstracción.** Este permite rellenar el hueco que existe entre los agentes con un alto nivel de abstracción (muy común en los agentes deliberativos) y el nivel básico de servicio. De esta manera se encapsulan los detalles de bajo nivel del despliegue para que los agentes no tengan que ser conscientes de ello. Un claro ejemplo son los servicios que ofrece JADE (Bellifemine et al., 2001) para intercambiar los mensajes entre agentes.
- **Nivel de interacción-mediación.** Este nivel ofrece soporte para regular el acceso a recursos compartidos y para mediar en la interacción entre los agentes. En esta última capa de soporte se está dando definitivamente una gran importancia al entorno ya que se convierte en una entidad activa, un árbitro, que ayuda a coordinarse a los agentes para que puedan explotar mejor el entorno y coordinar su comportamiento. Por ejemplo en este nivel se enmarcan servicios como los campos computacionales (Bandini et al., 2001; Mamei and Zambonelli, 2004) o las simulaciones de feromonas (Deneubourg et al., 1986).
- **Nivel de reflexión.** Este nivel todavía no está completamente aceptado, pero es bastante interesante. Consiste en dotar a los agentes cognitivos con la capacidad de modificar el comportamiento y funcionalidades del entorno. El trabajo con agentes artefactos (Ricci et al., 2002) entraría dentro de esta nueva categoría de servicio.

Estas tres (o cuatro) capas de nivel de soporte están apiladas, es decir, cada capa se basa en la anterior para poder ofrecer su funcionalidad. A su vez, los agentes pueden solicitar los servicios de cualquiera de estas capas, aunque por lo general cuanto más bajo sea el nivel de servicio que requieren menor es el nivel de abstracción que soportan, dificultando su portabilidad y reduciendo la robustez completa del MAS.

Una vez que se han definido algunos patrones de uso que los MAS tienen con el entorno es posible definir completamente el concepto de sistema multiagente situado (**S.M.A.S**). Como ya se ha comentado anteriormente en este trabajo, se define entorno como una abstracción de primer nivel que provee las condiciones necesarias para que los agentes puedan existir y mediar en la interacción entre agentes y el acceso a los recursos.

Esta definición se puede explotar en tres ideas concretas:

- Se recalca que el entorno es una abstracción de primera magnitud. Se convierte en una pieza independiente y fundamental que encapsula sus propias responsabilidades.
- También se hace hincapié en la condición de proveedor de condiciones idóneas para que los agentes existan. El entorno es parte del mundo en el cual los agentes interactúan, en el cual las acciones y los efectos de los agentes pueden ser observados y evaluados. Más aún, por si mismos, los agentes son sólo elementos que pueden realizar acciones, pero para construir un sistema realmente útil, estos agentes deben interactuar. El entorno les ofrece el nexo que conecta los agentes al sistema de trabajo.
- Por último, el entorno hace de mediador en la comunicación, ya sea agente-agente o agente-recurso. Esto significa que el entorno es una entidad activa en el reparto de responsabilidades de un **S.M.A.S**. Como mediador, el entorno no sólo permite la interacción, también la delimita y regula. De esta manera el entorno es un recurso muy valioso para los diseñadores de **S.M.A.S** ya que ofrece un nuevo espacio para plasmar las funcionalidades deseadas. Distinguir entre las responsabilidades de los agentes y del entorno mejora la encapsulación del **S.M.A.S** y ayuda a manejar sistemas muy complejos como aquellos que interaccionan con el mundo real. Esta encapsulación favorece la reusabilidad y el escalado del código, mejorando el proceso de producción del software.

#### 2.3.5.6 Responsabilidades del entorno

Una vez se ha descrito extensivamente el concepto de **S.M.A.S** se hace una revisión de todos los conceptos clásicos de los sistemas multiagente pero desde el nuevo enfoque de los **S.M.A.S**, centrándonos en el entorno y sus nuevas responsabilidades.

**El entorno como estructurador del sistema multiagente** Un entorno no es más que un espacio compartido por agentes, recursos y servicios. Todo ello, conjuntamente, conforma el sistema completo. Los recursos tienen un estado específico que puede ser manipulado por los agentes. Los servicios se consideran entidades reactivas que ofrecen cierta funcionalidad a los agentes. Por lo tanto los agentes, los recursos y los servicios están dinámicamente interrelacionado unos con otros. Ahora el papel del entorno es

definir las reglas que regirán esas relaciones. Por esa razón el entorno es ahora la entidad estructurante en el **S.M.A.S.**

Existen tres tipos de estructuramiento:

- **Estructura física.** Es decir, la estructura física, la topología, distribución, etc. Un ejemplo en (Bandini et al., 2001; Parunak, 1997).
- **Estructura de comunicaciones.** Transferencia de mensajes, agentes facilitadores, páginas amarillas y blancas (Bellifemine et al., 2001) o (Ricci et al., 2002).
- **Estructura social.** Organizando la estructura de los agentes, su jerarquía o sus roles como en (Odell et al., 2003; Omicini, 2001).

La estructuración es una parte fundamental del entorno. Las estructuras pueden ser fijadas por la propia naturaleza del dominio o pueden ser decisiones de diseño. Las estructuras físicas y de comunicaciones son una parte fundamental del contexto de despliegue y han de ser convenientemente abstraídas.

**El entorno como contenedor de recursos y servicios** Aunque parece obvio, hay que tener en consideración el hecho de que los recursos y los servicios normalmente tienen una estructura física marcada por la propia naturaleza de los mismos. El entorno debe contener y encapsular dichos elementos para dar una interface limpia, sencilla y general a los agentes. Esta política redundará en una mayor simpleza de programación de los agentes, lo que implica mejor software y mayores posibilidades de escalabilidad y portabilidad.

**El entorno como simulador dinámico** Más allá de la simple actividad de los agentes, el entorno puede tener procesos propios, independientes de los agentes. Estos procesos son necesarios para reflejar cambios en el entorno, ya sean derivados de los distintos procesos propios al **S.M.A.S** (por ejemplo feromonas digitales (Dorigo and Gambardella, 1997; Drogoul and Ferber, 1992; Parunak, 1997) o campos virtuales (Bandini et al., 2001, 2002) o para reflejar cambios en el mundo real que afectan al entorno. Dependiendo de esto las modificaciones del entorno afectan al soporte de nivel de abstracción o de interacción-mediación.

**El entorno es localmente observable por los agentes** Si las interioridades de los agentes deben ser invisibles para mantener el principio de encapsulamiento, el entorno, por contra, debe ser completamente observable por los agentes. Estos deben ser capaces de inspeccionar las diferentes estructuras del entorno, los recursos, servicios o la información publicada por otros agentes. Cabe destacar que esta observabilidad está directamente relacionada con la descripción semántica del sistema que puede ser definida con una ontología propia al entorno. Para los agentes cognitivos simbólicos esta es una característica fundamental del entorno con la que poder obtener información y deliberar sobre ella. Incluso para los reactivos la ontología diseñada sirve para definir su estructura implícita.

**El entorno es localmente accesible por los agentes** Al igual que para la observabilidad, acceder a la estructura está imitado al contexto en el que se encuentren los agentes. El acceso a la estructura espacial, en particular, es necesaria para soportar ciertas tareas que los agentes han de realizar para interactuar con el mundo real: mediciones, movilidad, etc. También el acceso a la infraestructura de comunicaciones ya sea directa (mensajes) como indirecta (marcas, campos, feromonas, etc.) debe ser accesible por los agentes.

En general, cualquier recurso puede ser percibido, modificado, generado o consumido por un agente. El control de qué agentes pueden acceder a qué recursos ha de ser responsabilidad del entorno y dependerá de factores de diseño o de funcionalidad. De igual manera que en la observabilidad, los agentes deben acceder a través del nivel de abstracción más alto posible.

**El entorno puede definir reglas para el sistema multiagente** Como se comentaba en el punto anterior, el acceso, ya sea sólo de lectura como de lectura y escritura debe ser controlado de alguna manera para evitar que agentes realicen acciones en contra de la política general del sistema. Este control se lleva a cabo en el entorno, y se basa en reglas, ya sean dinámicas o de diseño (estáticas). Normalmente el entorno será el garante de que el sistema en general se comporte de la mejor manera posible, sin permitir que ningún agente consuma demasiados recursos poniendo en peligro la consistencia del sistema.

**Modelo de referencia para el entorno** Una vez se han enumerado todas las posibles responsabilidades del entorno sólo queda presentar un modelo de referencia del cual se pueda partir a la hora de construir nuevos **S.M.A.S.** En particular, este modelo

describe una descomposición funcional de las aplicaciones del entorno, al menos en las funcionalidades que se diseñan «a mano» y no son producto de la evolución del sistema.

En el siguiente diagrama se muestra el esquema de dicho modelo. En la parte superior aparecen los agentes que toman decisiones y actúan en el entorno. En la parte inferior el contexto de despliegue es explotado por el entorno.

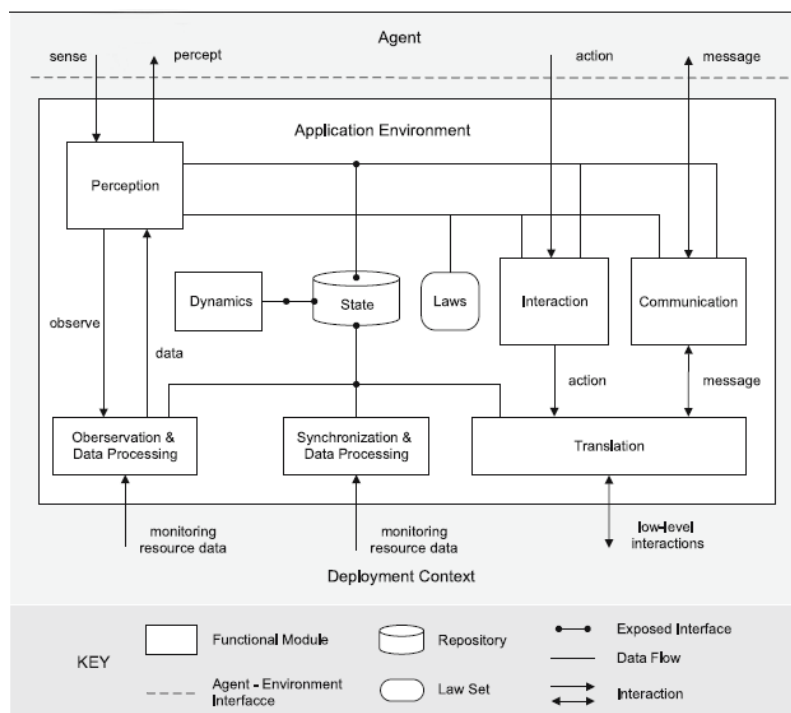


Figura 2.29: Estructura SMAS

Este modelo está formado por un conjunto de módulos que representan una funcionalidad concreta. Los agentes son capaces de «sensar» el entorno para obtener una «percepción» del mismo. También puede intercambiar mensajes con otros agentes. Por supuesto los agentes también pueden actuar sobre el entorno mediante acciones, tanto internas como externas.

- **Módulo de estado:** representa el estado actual del entorno. Esto incluye cualquier dato que sirva para caracterizar el estado actual del mismo: topología, datos sobre los agentes, etc. Este módulo publica un interfaz a los agentes para que éstos puedan acceder y modificar el estado del entorno en cada momento.
- **Módulo de sincronización y proceso de datos:** todos los procesos en los que estén involucrados sensores, calibraciones, lecturas, correcciones de datos,

interpolación o interpretación se encapsulan en este módulo. Además, si alguno de estos datos cambia, por ejemplo una cámara indica que hay un nuevo elemento en el entorno, el módulo se encarga de reflejar el nuevo estado en el módulo de estado.

- **Módulo de elementos dinámicos:** si un elemento dinámico del entorno provoca un cambio en el estado éste debe ser actualizado en el resto de módulos, por ejemplo una feromona o un temporizador.
- **Módulo de leyes:** todas las reglas que asocian los agentes con los servicios se gestionan en este módulo.
- **Módulo de percepción:** Cuando un agente «sensa» el entorno este módulo genera una percepción de acuerdo con el estado del entorno y las posibilidades sensoras del agente. Es decir genera una «instantánea» del entorno tal y como el agente debe verlo.
- **Módulo de observación y proceso de datos:** Este módulo procesa la información del contexto de despliegue y se la pasa al módulo de percepción, cumpliendo una misión similar al módulo de sincronización y proceso de datos, pero en este caso el destino de la información es el agente.
- **Módulo de interacción:** Este módulo maneja las acciones de los agentes en el entorno. El módulo de interacción encapsula el modelo de acciones que describen como las acciones que han sido ejecutadas por el agente son tenidas en cuenta.
- **Módulo de comunicación:** Este módulo hace las veces de servicio de mensajería, añadiéndole el control sobre la información que se transmiten los agentes entre sí. Por ejemplo, si una comunicación viola las leyes, el entorno las interceptará.

### 2.3.5.7 Niveles de soporte del modelo

Por último se describe la relación entre los módulos del modelo y los niveles de soporte tal y como se han descrito. El siguiente gráfico ilustra la distribución cartesiana en los ejes entorno y soporte:

- El nivel básico se encarga de la parte de monitorización que tienen los módulos «observación y proceso de datos» y «sincronización y proceso de datos».

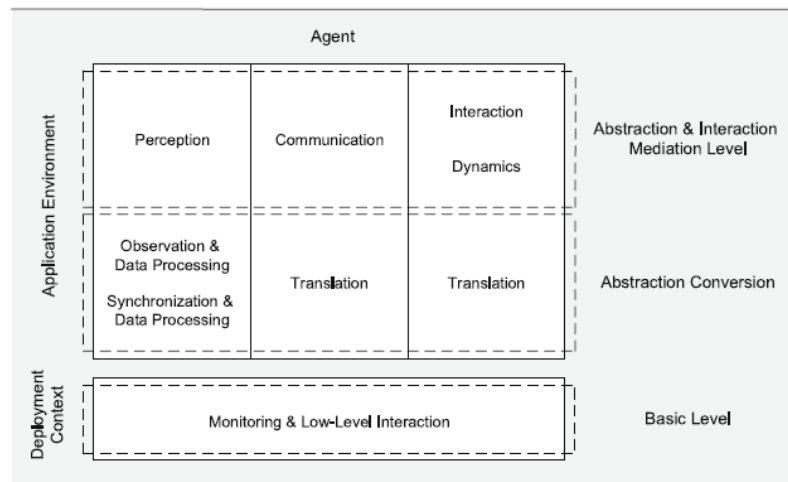


Figura 2.30: Niveles de soporte del modelo

- El nivel de abstracción se encarga claramente de parte de los módulos «observación y proceso de datos» y «sincronización y proceso de datos», así como de las traducciones de los datos.
- El nivel de interacción y mediación se encarga de «percepción», «comunicación», «interacción» y «procesos dinámicos»

## 2.3.6 Casos prácticos de MAS Situados

### 2.3.6.1 Introducción

En este apartado vamos a revisar distintos y variados casos prácticos relacionados de una u otra manera con **Sistema Multi Agente Situado** en **Entornos Virtuales Inmersivos**. El objetivo de este capítulo es dar una visión general del estado del arte en dos ramas que, aunque tienen varios puntos en común, ha tenido caminos muy separados. Por lo tanto comenzaremos por los ejemplos más teóricos y relacionados con la definición formal de Weyns para ir poco a poco acercándonos a las experiencias en entornos virtuales inmersivos como *Second Life* y *Open Simulator*.

### 2.3.7 Modelo de Weyns, Schelfthout y Holvoet, 2005

Puede que el caso mejor y más profusamente documentado de un sistema multiagente situado (**S.M.A.S**) sea el de control de una factoría automatizada con vehículos de guiado automático (AGV). Dicho sistema ha sido documentado en (Viroli et al., 2005; Weyns and Holvoet, 2004; Weyns et al., 2005, 2007b).

Un AGV es un sistema de transporte que usa vehículos no tripulados para transportar cargas dentro de un almacén. Las aplicaciones típicas son reempaquetar y distribuir bienes entrantes a las diferentes ramas, así organizar los productos manufacturados en sus distintos lugares dentro del almacén. Un AGV utiliza una batería como fuente de energía y pueden moverse autónomamente a lo largo de todo el almacén, siguiendo un camino físico en el suelo de la factoría, guiado por un sistema de navegación laser. Un AGV puede recoger una carga en una determinada localización y dejarla en otra. Así mismo puede aparcar en lugares prefijados a tal efecto cuando está ocioso y cargar su batería en un punto de recarga.

#### 2.3.7.1 Funcionalidades

El objetivo principal del sistema es mover cargas de un lugar a otro. Los requerimientos de transporte son originados por actores externos, ya sea software logístico, empleados u operadores. Un transporte está formado por un conjunto de tareas simples. Las tareas más importantes son:



- Asignación: carga, lugar de origen, lugar de destino
- Enrutado: los AGVs deben moverse por el camino más eficiente
- Recolección de información de tráfico: para poder calcular el enrutado de manera óptima
- Evitar colisiones: es importante que los AGVs no colisionen
- Prevención de bloqueos: dado que los AGV sólo pueden pasar por su camino es importante evitar situaciones de interbloqueo entre ellos
- Cuando un AGV está desocupado puede aparcar en el parking habilitado para ello. Cuando tiene poca energía debe dirigirse al punto de recarga.

Aunque en un primer momento se pensó enfocar el problema con un sistema centralizado se encontraron importantes problemas que sugerían una solución descentralizada. Este cambio estaba dirigido a aumentar la autonomía, flexibilidad y apertura del sistema. De esta manera los vehículos se tornaban en agentes independientes capaces de tomar decisiones por sí mismos y de sincronizarse de manera automática.

El problema se abordó, obviamente, con un modelo **S.M.A.S**. El **S.M.A.S** se compone de dos tipos de agentes: los «Agentes de Transporte» y los «Agentes de AGV». El primero representa una carga que debe ser transportada y el segundo un vehículo AGV. Las comunicaciones se realizan a través de un sistema inalámbrico. Los vehículos AGV están situados en el entorno físico, pero este entorno está muy limitado: los vehículos no pueden manipular el entorno excepto cogiendo o soltando cargas. Para simular este entorno físico se implementó un entorno virtualizado réplica del real. En este caso cada AGV es un elemento computacional distinto cuyo único vínculo con el resto es una conexión inalámbrica (y por lo tanto poco fiable), esto significa que cada vehículo debe tener su propia réplica de entorno virtual local que se actualice automáticamente. Dichos entornos virtuales locales se fusionan generando un único entorno virtual distribuido con la información sensorial de todos los elementos.

### 2.3.7.2 Distribución de las responsabilidades

**Asignación de transportes** . Los transportes están representados por los agentes de transporte por cada nuevo transporte se genera un nuevo agente de transporte que

es el responsable de asignar ese transporte al AGV y asegurarse que es completado correctamente. Los agentes de transporte utilizan su entorno virtual para localizar a los agentes AGV. Si un AGV es incapaz de completar la solicitud derivará la petición a otro.

**Enrutado** . El entorno virtual dispone de un mapa de todo el almacén, mientras que el software de navegación de los AGV es capaz de reconocer ciertas marcas de posicionamiento que le ayudan a saber dónde se encuentra. El mapa está construido como un grafo y cada segmento tiene un coste proporcional al tiempo que el AGV tarda en recorrerlo. Una vez tomadas en consideración estas variables se define el camino con el menor peso, aunque el AGV siempre puede tomar otra ruta en cualquier intersección en base al estado actual del sistema.

**Recolección de información de tráfico** . Aunque la ruta marcada sea óptima en un momento dado no significa que al instante siguiente lo siga siendo. Esto implica estar constantemente monitorizando el estado y la posición de los AGV y actualizar el mapa virtual que tiene el entorno para que, en caso de ser necesario, los Agentes AGV indiquen otro camino óptimo a los AGV en vuelo.

**Prevención de colisiones** . Los Agentes AGV son capaces de prever las colisiones coordinándose a través del entorno virtual. Utilizando una virtualización del contorno del AGV denominado «hull» se discretiza el espacio de los AGV no permitiendo que dos «hulls» ocupen el mismo sitio en el mismo momento.

**Prevención de bloqueos** : En este caso el enfoque distribuido resulta peor ya que la solución descentralizada siempre puede encontrar casos en los que falle. Para solucionarlo se desarrollaron agentes especializados a tal efecto.

### 2.3.7.3 Descripción del entorno virtual de los agentes AGV

En este diagrama se representa el modelo del entorno siguiendo el esquema planteado en la teoría, pero identificando los elementos diferenciadores del problema.

Como se puede observar el módulo de estado se ha dividido en tres partes.

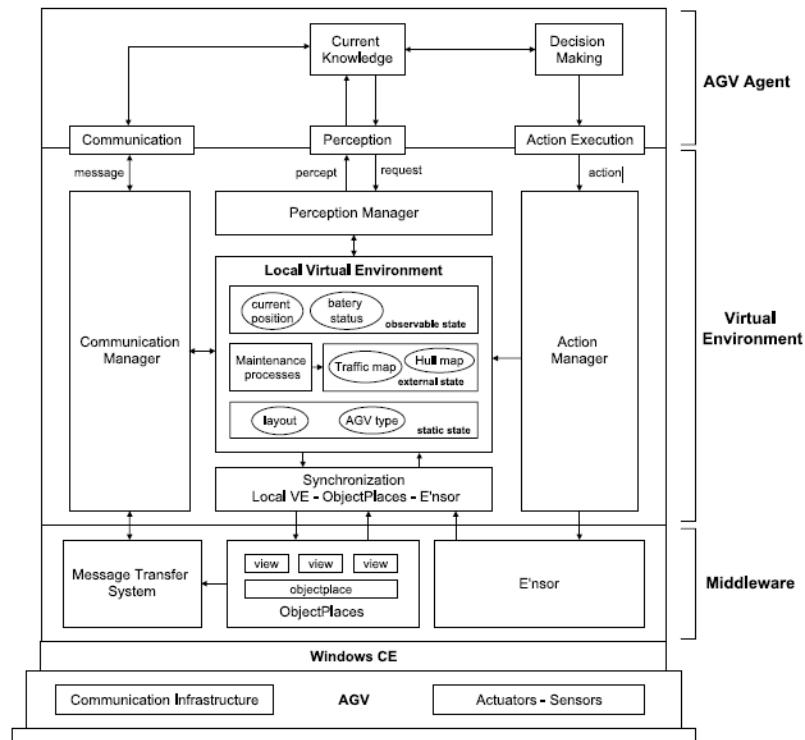


Figura 2.31: Esquema entorno AGV

- **Estado estático.** Elementos que no cambian con el tiempo como el mapa de la factoría.
- **Estado observable.** Estados que todos pueden leer pero sólo uno puede escribir. Por ejemplo la posición de los AGV. En este caso no es posible el conflicto.
- **Estado compartido.** Estados que todos pueden leer y escribir. En este caso los conflictos se deben resolver dependiendo del tipo de consulta.

Para mantener el estado consistente, el entorno virtual debe estar realizando constantemente tres actividades:

- Sincronizar el estado del entorno virtual local con los sensores de los AGVs con valores como la posición.
- Sincronizar el estado del entorno virtual local con otros AGVs a través de un middleware que se encarga de replicar la información en forma de tuplas.
- Mantener el estado del entorno virtual local en función de los procesos dinámicos.

Por último, existen tres módulos independientes: percepción, actuación y comunicación. Estos tres módulos ofrecen primitivas de alto nivel a los agentes.

#### 2.3.7.4 Conclusiones

Hasta el momento de publicación del estudio con este sistema, se había implementado el enrutado, el intercambio de información y la prevención de colisiones, así como validado el modelo en dos AGV físicos. Los resultados hasta el momento no son completamente concluyentes ya que el sistema no está completo.

Los siguientes pasos son: implementar completamente la prevención de bloqueos y ampliar las pruebas a seis AGVs.

### 2.3.8 Experiencia de Bandini, Manzoni, y Simone, 2001

En (Bandini et al., 2001) se presenta una aplicación de los MAS Multicapa (MMASS) que aprovecha las características espaciales y territoriales para el cálculo de posición óptima para la construcción de centros comerciales en el extrarradio de las ciudades. Se ha demostrado (Coclelis, 1997) que las simulaciones espaciales son muy útiles para la planificación urbana ya que los numerosos factores como accesos, distribución geográfica, zonas residenciales, etc. conforman un sistema realmente complejo. Para este problema se utiliza la herramienta de trabajo MABS (Bandini et al., 2001) y se plantean dos aproximaciones distintas: **S.M.A.S** territorial y **S.M.A.S** estratégico.

Para el **S.M.A.S** territorial únicamente se tiene en cuenta la relación geográfica de las posibles ubicaciones de los centros comerciales y se genera una rejilla que divide el terreno en casillas cuadradas. Cada una de las rejillas puede ser de cinco tipos: residencial, industrial, no construible, carretera y disponible. Estos cinco son los posibles tipos de agente. Los cuatro primeros sólo pueden tener un estado, pero el último, los «disponibles» pueden tener tres posibles valores: vacío, pequeño y grande.

Por definición, todos los agentes emiten dos campos y según su tipo. Los distintos valores del campo atraerán o repelerán agentes del tipo «disponible». Dado que estamos hablando de un modelo espacial, es importante definir la distancia. El modelo se basa en un grafo no dirigido de sitios, se define la distancia como el número de saltos de un sitio a otro, con un máximo de 3. La tabla de valores de campo que se utilizó es la siguiente:

	Distancia 1	Distancia 2	Distancia 3
Residencial	9	9	1
Industrial	1	2	0
Protegido	-6	2	-2
Carretera	6	-1	-10
Disp. Grande	-9	-7	-1
Disp. Pequeño	-8	-4	2
Vacio	3	-8	-8

Figura 2.32: Tabla de distancia

El **S.M.A.S** estratégico se construye como un conjunto de **S.M.A.S** territoriales para poder replicar el esquema a una escala mayor. Además incluye el concepto de competición. Una vez generados todos los agentes del modelo completo dichos agentes compiten tomando alternativamente roles agresivo y defensivo. Aquellos agentes que mejores características poseen resultan ganadores y aumentan las mismas, y viceversa. Finalmente, la mejor posición para ubicar un centro comercial estará señalada por el agente más fuerte.

### 2.3.9 Propuesta de Helleboogh, Vizzarri, Uhrmacher y Michel, 2007

En (Helleboogh et al., 2007) se plantea el hecho de que los sistemas multiagente que tienen que manejar sistemas en el mundo real son inherentemente complejos y muy dinámicos. Esto implica la necesidad de disponer de un modelo que incluya este dinamismo como una parte muy importante del entorno simulado.

En su trabajo, Helleboogh et al., proponen un entorno simulado que responde a un formalismo propuesto. Mediante dicho formalismo es posible estructurar el simulador del entorno y modelar su implementación. Dicho formalismo tiene las siguientes funciones:

- Representar de una manera explícita el dinamismo que ocurre en el entorno
- Especificar cómo el dinamismo del entorno afecta a los agentes
- Especificar cómo se genera, interfiere y se termina dicho dinamismo en el entorno

El esquema propuesto se descompone en dos diferentes conceptos:

- Entidades del entorno, que representan entidades caracterizadas por su propia y diferenciada existencia en el mundo real.
- Propiedades del entorno, que representa una cantidad distribuida que representa una característica medible y común a todo el entorno real.

Independientemente de su tipo, cada entidad tendrá asignado un estado en cada momento, dentro de un posible conjunto de estados inicial. Para especificar el estado inicial de cada entidad se define una función «init» que mapea a cada entidad su estado inicial.

También hay que relacionar los agentes y el entorno. Para ello es fundamental definir que los Agentes no son elementos externos al entorno, más al contrario, son parte del mismo. A este concepto se le denomina «personalización» (embodiment). Por lo tanto los agentes se personalizan como «entidades del entorno» a través de un cuerpo físico particular que existe en la realidad.

Por último, para que el formalismo sea completo, es necesario describir los mecanismos necesarios para modelar el dinamismo del entorno. Esto se plantea mediante el concepto de actividad. Una actividad tiene tres componentes:

- el elemento del entorno (ya sea entidad o propiedad) al que afecta
- el intervalo de tiempo en el que ocurre
- la estrategia de evolución, que define la manera en la que el estado del elemento va evolucionando con el tiempo

Una vez se ha definido el modelo se aplica a un conocido problema: el RoboCup. En el RoboCup Soccer un conjunto de robots juegan un partido de fútbol. Tienen sus porterías, campo, balones y por supuesto los robots. Para el ejemplo únicamente tienen en cuenta dos robots y una jugada.

Según el modelo anteriormente enunciado, las entidades del entorno son: dos robots, una pelota, un campo y una portería. Las propiedades del entorno son: temperatura y humedad. Los agentes personalizados son el robot Bob y el robot Alice.

Los resultados, aunque puramente teóricos indican que tiene sentido el tratamiento de un sistema MAS específicamente situado en un problema físico mediante el diseño de un entorno virtual en el cual los agentes pueden desenvolverse.

### 2.3.10 Modelo teórico de Wang et al., 2011

En (Wang et al., 2011) se plantea un modelo teórico para sistemas multiagente basados en el aprendizaje motivado. Además muestra dos casos prácticos que ilustran perfectamente el potencial de los entornos virtuales en la construcción de sistemas multiagente.

En el primer caso práctico se describe el uso de la herramienta MR-Collab con este entorno: dos salas separadas se configuran como entornos inteligentes. El sistema no solamente funciona como una plataforma colaborativa para usuarios distribuidos, sino que además es capaz de aprender patrones que mejoran el diseño de las actividades.

En los entornos inteligentes existe un proyector, algunos objetos y algunos marcadores. Una vez se sitúa un individuo con un casco de realidad aumentada (head-mounted display HMD) los objetos del entorno virtual aparecen representados sobre su visión del entorno.

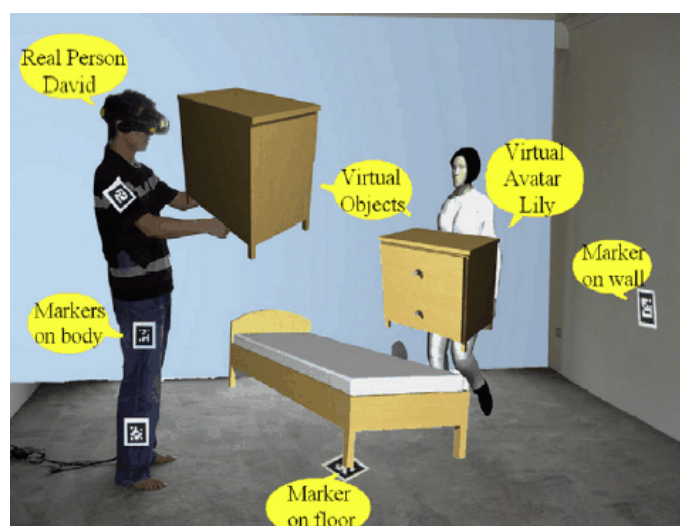


Figura 2.33: Ejemplo entorno inteligente

Mediante estos dos entornos inteligentes, los dos diseñadores pueden colaborar en su trabajo ya que ven e interactúan como si estuvieran físicamente en el mismo entorno. Pero se da un paso más en el estudio: se incluye un sistema multiagente que gestiona las interacciones virtuales entre las dos salas. Los agentes modifican los sensores del entorno físico, generando los efectos físicos esperados. Por ejemplo, puede tener en cuenta el número de personas que están en el entorno y generar las sillas necesarias.

En el segundo caso se estudia la herramienta VR-Collab. Este sistema es una extensión del primero y extiende el modelo de sólo dos salas a múltiples salas. Pero lo más innovador

del sistema es que en lugar de utilizar salas ad-hoc utilizan un «proveedor de servicio» de simulación física inmersiva multiusuario como es *Second Life*.

Esta herramienta parte de la misma base: se mantiene un espacio virtualizado y unos objetos que también existen en ese entorno virtual, aunque desaparecen los sensores y las cámaras ya que no hay un entorno inteligente en el mundo real. El desarrollo de la herramienta VR-Collab se ha llevado a cabo en lenguaje LSL, el estándar de programación script dentro de *Second Life*.

Como conclusión interesante, este trabajo introduce los conceptos de entornos inteligentes y de agentes basados en modelos. Sin embargo los sistemas enunciados no han sido completamente diseñados ni se tienen experiencias reales sobre su uso.

### **2.3.11 Planteamiento de Ritchings, Anastassakis y Panayiotopoulos, 2001**

En su trabajo llamado Sistemas multiagente como entornos virtuales inteligentes (Ritchings et al., 2001) se plantea fusionar los sistemas multiagente y los entornos virtuales inteligentes. Esta fusión se realiza mediante la herramienta VITAL, un sistema inteligente multiagente capaz de soportar aplicaciones con entornos virtuales inteligentes multipropósito.

En primer lugar se maneja el término Entorno Virtual Inteligente (IVE) como acrónimo para definir un entorno virtual con una capacidad de proceso inherente, pero sin llegar a definir más detalles.

VITAL significa «Virtual InTelligent Agents with Logic» y es una simulación de un laberinto. Los agentes deben explorar un laberinto dado, localizar elementos específicos dentro de él y procesarlos en función de las instrucciones de los usuarios (por ejemplo mover dichos objetos). El sistema monitoriza utilizando un visualizador 3D de visualización completa.

El sistema distingue entre tres tipos diferenciados de elementos discretos: mundos, agentes y visualizadores. Cada módulo está implementado desde el modelo cliente-servidor, de tal manera que el mundo actúa como servidor y los agentes y visualizadores como clientes siguiendo este esquema:

El servidor del «mundo» hace las veces de entorno virtual común, utilizando un modelo de orientación a objetos para representar todos los elementos del entorno.



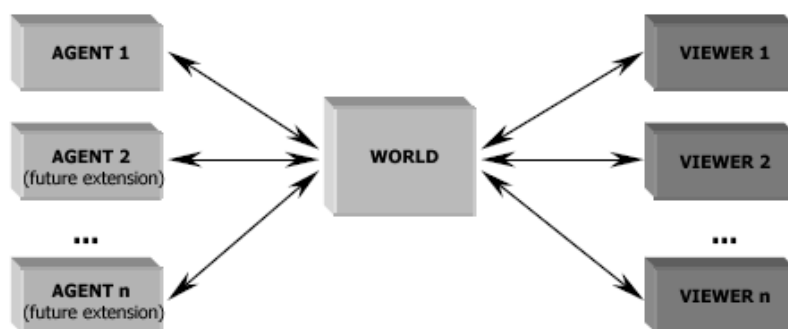


Figura 2.34: Esquema cliente servidor

Los visualizadores realizan tareas únicamente de visualización y control de los agentes a través de un interfaz 3D. Dicho interfaz se genera a través del estándar VRML (Virtual Reality Modelling Language).

Los agentes utilizan un modelo BDI basado en lógica de predicados para planificar sus acciones. Este es un ejemplo:

```
{
beliefs({ holding({}) }),
desires({ explore(X, Y), pickup(theRedBall),
drop(theRedBall, 1, 2), pickup(theBlueBall),
drop(theBlueBall, 3, 2), move(2, 2) }),
intentions({})
}
ability: move(X, Y)
preconditions: {at(CX, CY), connects(CX, CY, X, Y)}
delete: {at(CX, CY)}
add: {at(X, Y)}
```

Como conclusión general se reconoce el gran valor de la herramienta desde el campo de la investigación, aunque es obvio que el modelo estaba en una fase muy temprana de desarrollo.

### 2.3.12 Experimento de Mamei y Zambonelli, 2004

En su trabajo (Mamei and Zambonelli, 2004) sobre la coordinación del movimiento de los agentes utilizan un entorno virtual inmersivo denominado Quake 3 Arena. Dicho motor

gráfico está extraído del juego Quake Arena de IDSoftware y permite hacer que algunos de los jugadores sean controlados por bots completamente programables, a la vez que nos permite entrar al mismo escenario mediante un entorno virtual 3D interactivo.

Aunque el trabajo no se basa en ningún modelo de sistema multiagente situado tiene componentes que remiten claramente a estos esquemas. En dicho estudio se plantea la necesidad de disponer de herramientas para la coordinación de los agentes que se encargan de manejar los jugadores, también denominados simplemente «bots».

El concepto que se usa en esta experiencia es el de campos, denominándolo CoFields como contracción de «Computational Fields» siguiendo el mismo principio que se presentaba en (Bandini et al., 2002). Los campos son abstracciones matemáticas del mismo concepto físico de campo gravitacional o campo electromagnético: una propiedad específica de un punto determinado en el espacio que es detectable por todos los agentes. No sólo debe ser detectable, sino que en ocasiones también es modificable y acumulable

En el presente caso de estudio se han definido los campos PRESENCE y FLOCK:

- El campo PRESENCE indica la presencia de un jugador (bot) en el entorno virtual. El objetivo es poder realizar un movimiento de agrupación de los bots. Cada agente «emite» su propio campo pudiendo calcular el campo total como la función sumatoria de todas las presencias de todos los agentes. En ocasiones, dependiendo de si interesa o no contar con el campo del propio agente, se elimina de ese sumatorio la propia PRESENCE del agente que lo calcula.
- El campo FLOCK se usa para otro fin: la formación. Si con PRESENCE se conseguía que los agentes se juntaran en un único lugar, aún quedaba otro problema organizativo ya que los agentes tendían a «amontonarse» en un sólo lugar. Para solucionar este problema se utilizó el campo FLOCK, una función sinoidal periódica que asigna valles y colinas regulares consiguiendo de esta manera que los agentes se alineasen regularmente en el espacio, en lugar de formando una agrupación caótica.

Como conclusión, aunque este trabajo es relativo a un prototipo y se han descubierto numerosas mejoras a implementar, el propósito inicial de este proyecto, comprobar la viabilidad del uso de campos (CoFields) se ha cumplido.

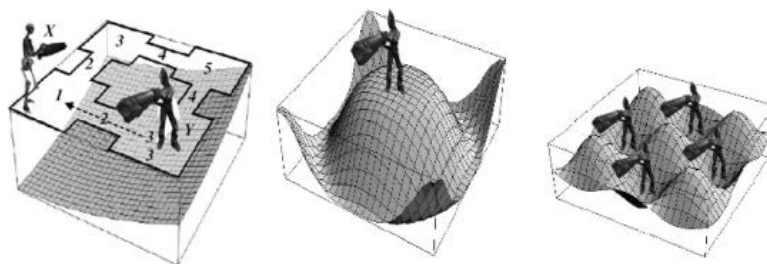


Figura 2.35: Ejemplo CoFields

### 2.3.13 Estudio de Friedman, Steed y Slater, 2007

En su trabajo (Friedman et al., 2007) entran de lleno a estudiar un *Entorno Virtual Inmersivo* multiusuario: *Second Life*. En este caso la investigación no se centra tanto en los aspectos de sistema multiagente sino en la faceta social de este metaverso. Sin embargo la métrica que se utiliza para realizar el estudio sí que es propia de las técnicas típicas de **S.M.A.S**: los campos, aunque en el trabajo no se mencione este concepto. También existe otro nexo en común con las técnicas multiagente: el software utilizado para realizar las mediciones.

El experimento que se describe en el artículo tiene como objetivo estudiar las relaciones espaciales humanas dentro de un metaverso, es decir la proxémica virtual. Utilizando agentes muy simples (objetos con código LSL) se realizan mediciones de la posición de los avatares en una región.

Se definen los siguientes valores de proximidad basándose en los estudios sociológicos:

- íntimo:  $< 0.5$  m
- personal:  $0.5 < x < 1.5$  m
- social:  $1.5 < x < 4$  m
- público:  $> 4$  m

Las mediciones del trabajo son algo escasas, por lo que las conclusiones no son completamente claras, pero parece obvio que existe cierto modelo de «campo» subyacente similar al mostrado en (Mamei and Zambonelli, 2004) con el campo PRESENCE.

### 2.3.14 Aproximación de Cranefield y Li, 2010

En (Cranefield and Li, 2010) (Cranefield, et al., 2010) Cranefield y Li plantean el uso y adaptación de herramientas software propias de los MAS para el estudio de las interacciones sociales humanas en un entorno virtual inmersivo multiusuario o metaverso, en particular en *Second Life*.

Los autores argumentan que, a sabiendas de que los MAS han sido inspirados por los esquemas sociales humanos, es factible utilizar los esquemas propios de los MAS para investigar de vuelta las relaciones sociales humanas dentro de entornos virtuales. Algunos ejemplos son los agentes capaces de manejar conceptos como confianza, reputación, expectativas, compromiso o narrativa.

Para llevar a cabo el experimento los autores han desarrollado con objeto de tipo HUD virtual que es anexo al visor de los avatares y es capaz de interactuar con un monitor de comportamientos sociales esperados mediante un interfaz mixto http+Twitter y xmlRpc+Java.

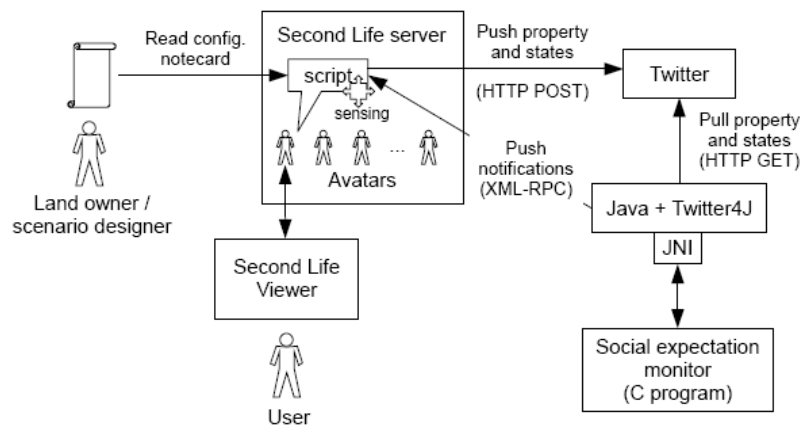


Figura 2.36: Esquema del modelo Cranefield

A través del monitor de comportamientos sociales y basándose en un conjunto de reglas en lógica temporal, el sistema es capaz de indicar las violaciones y los cumplimientos de dichas reglas por los avatares que en ese momento están compartiendo entorno virtual en *Second Life*.

Aunque el sistema es técnicamente viable, los autores indican que el «lag» del sistema hace bastante complicada la monitorización efectiva. Sin embargo el modelo ha sido probado con éxito en escenarios pequeños y con un grupo muy reducido de avatares.

### 2.3.15 Resumen del estado del arte en Sistemas Multiagente Situados

Este estado del arte ha puesto de manifiesto la creciente importancia de los sistemas multiagente situados y los avances investigadores que se han realizado en este sentido. También se han expuesto distintos ejemplos de la evolución de los diferentes tipos de metaversos, entornos virtuales y simuladores. Dicha evolución todavía no tiene un cauce bien definido.

En ninguno de los estudios analizados ambas corrientes de investigación y desarrollo están claramente en contacto. De hecho, hasta donde se ha podido investigar, no existe un modelo claro de **S.M.A.S** que esté ubicado en un **V.I.E** y aproveche las ventajas del mismo. Sin embargo la mayoría de las ideas están ahí con un gran potencial para explotarlas.

Algunos de los conceptos que no se han estudiado suficientemente son: persistencia, modelo de procesos y de datos subyacentes, principio de localidad, etc. Tampoco se ha encontrado un framework de trabajo suficientemente evolucionado como para poder utilizarlo en proyectos multiagente. Estas carencias encajan perfectamente con las identificadas en el capítulo 1.

## 2.4 Estado del Arte relativo a los objetivos

Ya hemos revisado el estado del arte general tanto de la *Realidad Virtual* como de los *Sistema Multi Agente Situado* en *Entornos Virtuales Inmersivos*. Partiendo de esta base, en esta sección vamos a estudiar cuáles de los objetivos e hipótesis definidos en los puntos 1.2.2 y 1.3.2.

Tal y como se ha descrito en la tabla 1.2, los problemas relativos a la implantación de la **V.R** están asociados a ciertas carencias de elementos de toda índole: desde falta de modelos teóricos a la inexistencia de software disponible para la implantación de proyectos, pasando por falta de metodologías o arquitecturas específicas.

Si bien es cierto que estas carencias detectadas implican que no existen elementos idóneos para cubrir una necesidad, es interesante indagar sobre qué otros elementos (teóricos o prácticos) son similares o puedan ser tomados como base sobre la cual construir nuestra propuesta.

Para llevar a cabo este estudio vamos a tomar el conjunto de necesidades y carencias detectadas y repasar una por una qué existe actualmente y cómo podremos argumentar nuestra propuesta en los siguientes capítulos de esta tesis.

### 2.4.1 Metáfora antropomórfica adecuada

Si hablamos de metáfora antropomórfica aplicada al diseño software y en especial a la interacción humano-máquina (Helander, 2014) puede que la más importante y la más extensamente utilizada sea la «**metáfora del escritorio**»

La metáfora del escritorio y la metáfora del papel fueron enunciadas a principios de la década de 1970 y puestas en marcha en la siguiente década, siguiendo hoy en día en plena validez. El escritorio, el papel, los lápices, etc. son todos elementos propios de un puesto de trabajo en un *entorno* de oficina. Esta relación incluso dio nombre a un subconjunto de aplicaciones: «ofimática». La penetración de la ofimática en el entorno de oficina es, al menos en los países industrializados, tan alta que imaginar un puesto de oficina sin un PC o similar nos resulta chocante. Este éxito en la «digitalización» de un entorno de oficina significa que se ha demostrado, desde el punto de del rendimiento económico y laboral, que el papel y la calculadora «virtualizados» son superiores al clásico.



Figura 2.37: Metáfora y su tecnología de soporte

Pero no podemos olvidar otros importantes actores en este caso de éxito: la pantalla, el teclado y el ratón. A estos elementos se les denomina *tecnología de soporte* y se puede generalizar su definición como aquellos elementos tecnológicos hardware sin los cuales un paradigma o metáfora no se pueden implementar.

Resulta interesante observar que cuando Alan Key enunció la citada *metáfora de escritorio* no existía el hardware necesario para implementarla. No fue hasta 1984 con el Apple Macintosh (Williams, 1984) que se existió un equipo capaz de satisfacer la metáfora de una manera mínimamente aceptable. Es decir, aunque la metáfora ya había sido enunciada y ya se había definido la estrategia a seguir, hasta que no existió el hardware que permitiera desarrollarla no se pudo demostrar. Eso, sin embargo, no le resta importancia ni perjudica su validez, más al contrario, una vez han pasado más de treinta años podemos dar dicho enfoque como un éxito.

Distintas conclusiones pueden ser extraídas de esta reflexión, pero sobre todo vamos a centrarnos dos:

- Primera: usar una metáfora intuitiva y antropomórficamente consistente parece una buena estrategia.
- Segunda: aunque la tecnología hardware sea inmadura o incluso inexistente, no implica que la metáfora sea errónea

Volviendo al campo de aplicación de la **V.R.**, parece una buena estrategia tomar el «paradigma del escritorio» y ampliarlo como veremos al describir mi propuesta en el punto 1.4.1.

## 2.4.2 Principios de diseño

Aunque no existe un conjunto de principios de diseño específicamente definido para la elaboración de **V.I.E's**, es interesante la propuesta de (Zhao and Moere, 2008). Aunque

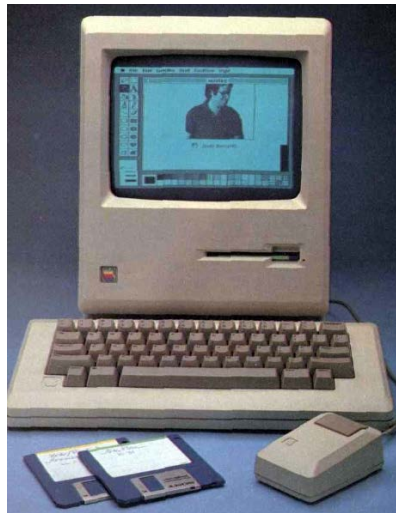


Figura 2.38: Apple Macintosh: Fue el primer computador personal que se comercializó con éxito que usaba una interfaz gráfica de usuario y un ratón

en realidad estos principios están orientados a la representación de información de manera visual, estos tres principios que se describen a continuación pueden ser más o menos transportados a nuestro problema concreto:

1. **Imagen mental única.** Cada elemento representativo debe corresponder a una imagen mental concreta y única. Por ejemplo «silla» es un elemento típico y comprensible con una función concreta, sin embargo «mueble» no tiene asociada una imagen mental única.
2. **Asequibilidad.** Cada elemento debe servir como «motivador» y como «referente». Es decir, cada elemento debe ser capaz de despertar una idea de uso y una idea de contexto. Por ejemplo una cuerda invita a tirar de ella para desencadenar una acción, pero también permite intuir que es parte de un todo al que está atada.
3. **Intuitividad.** Los elementos que se construyan deben ser cosas con las que la gente esté familiarizada y comprenda su significado de manera intuitiva sin tener que realizar una formación o entrenamiento previo. Por ejemplo, las personas están familiarizadas con la metáfora «plantas verdes» y son comprensibles sin mayor necesidad de conocimiento, pero por ejemplo «motor a reacción» sólo es intuitivo si se ha estudiado el tema.

De manera general, y siendo un concepto extrapolable a cualquier metáfora derivada de la «metáfora del escritorio» expuesta anteriormente, *a menor esfuerzo que cualquier*



*persona debe utilizar para comprender un concepto, mayor éxito a la hora de utilizar dicho concepto.*

Basándonos en esta idea se propondrá un conjunto de principios de diseño tal y como se indica en el punto 1.4.2.

### 2.4.3 Formalización abstracta

Como se verá en la propuesta de metáfora antropomórfica en el punto 1.4.1 y también en 1.4.2, se propondrá un elemento concreto como concepto clave y principal de la metáfora: **el entorno**. De hecho la metáfora se denominará «metáfora del entorno».

Es por tanto necesario definir este singular concepto conocido como **entorno** de una manera abstracta y concreta, que no deje lugar a dudas y a ser posible de manera matemática.

Indagando en este sentido existen tres ramas que podrán ser aprovechadas como fuente de conocimiento:

1. **El entorno como elemento matemático**, en (Bourbaki, 1998; Runde, 2006) el «entorno» o vecindad está íntimamente ligado al *espacio topológico*. En este contexto se define entorno como un conjunto de posiciones accesibles entre ellas, es decir, aquel espacio que exhibe continuidad y que permite la existencia de al menos un posible camino entre dos puntos A y B cualesquiera.
2. **El entorno como mediador de Sistemas Multiagente**, ya hemos estudiado en el punto 2.3.6, existen numerosas publicaciones como (Bellifemine et al., 2001), (Ferber, 1999), (Mohan and Laird, 2009), (Bandini et al., 2002), (Bandini et al., 2001; Platon et al., 2007), (Bandini et al., 2001; Mamei and Zambonelli, 2004), (Ricci et al., 2002), (Bandini et al., 2001; Parunak, 1997), (Bellifemine et al., 2001), (Odell et al., 2003; Omicini, 2001), (Dorigo and Gambardella, 1997; Drogoul and Ferber, 1992; Parunak, 1997), (Viroli et al., 2005; Weyns and Holvoet, 2004; Weyns et al., 2005, 2007b), (Bandini et al., 2001), (Couclelis, 1997), (Bandini et al., 2001), (Wang et al., 2011), (Ritchings et al., 2001), (Mamei and Zambonelli, 2004), (Friedman et al., 2007), (Mamei and Zambonelli, 2004) o (Cranefield and Li, 2010) el entorno siempre está presente con mayor o menor importancia en todos estos estudios. Es sin duda un buen punto de partida.

3. **El entorno según Wooldridge**, aunque en (Wooldridge, 2002; Wooldridge and Jennings, 1995) nunca se define el entorno bajo la perspectiva que estamos proponiendo y dándole la importancia que se merece, sí que es cierto que en el capítulo 2 de (Wooldridge, 2002) se define un álgebra relativa al funcionamiento de los **M.A.S** que puede servir como base matemática

Sin embargo ninguno de estos enfoques era suficientemente potente como para soportar las definiciones formales que necesitaremos en mi propuesta descrita en los puntos 1.4.4 y 1.4.5. Es por tanto necesario una formalización más eficaz y por esa razón se propone lo descrito en el punto 1.4.3.

#### 2.4.4 Método formal para medir inmersión

Aunque existe alguna publicación que ha indagado ligeramente sobre este concepto como en (Slater et al., 2009), no existe un método cuantitativo que permita evaluar un «entorno virtual». En otras palabras, es posible decir «este entorno virtual es inmersivo», pero no es posible indicar «cuánto de inmersivo es», y en realidad si comparásemos dos entornos virtuales según su cualidad inmersiva estaríamos más abriendo un debate subjetivo que una afirmación objetiva.

Mi propuesta en este sentido se apoyará en el concepto teórico abstracto de entorno y se ilustra en el punto 1.4.4.

#### 2.4.5 Método formal para medir validez ecológica

La «validez ecológica» es un concepto común en ciencias médicas y fisiología humana, imbricada fuertemente en el antropomorfismo inherente a estas disciplinas. Es también, profusamente comentada en numerosos estudios sobre **V.R** tal y como se ha descrito en 2.2.9 en (Rothbaum et al., 2014), (Kim et al., 2005), (Tarr and Warren, 2002), (Rizzo and Buckwalter, 1997), (Rizzo et al., 2004) o (Rizzo et al., 2006).

Sin embargo estamos en el mismo caso que la «inmersión», no es posible por ahora realizar una medición formal sobre este concepto. Mi propuesta también se sustenta sobre el modelo teórico abstracto de entorno y se definirá en el punto 1.4.5.

### 2.4.6 Tipos de elementos y servicios posibles

Una taxonomía bien definida y estructurada es fundamental para sustentar tanto un modelo teórico como una metodología, sin embargo no existen hasta el momento estudios que definan una tipología similar. Es sin embargo necesario citar a (Milgram et al., 1995) que ya en 1995 fueron capaces de intuir que la distinción entre **V.R** y **A.R** no era más que diferentes puntos de un mismo camino. Siguiendo el ejemplo de estos autores e inspirado por este principio de simplicidad propongo la taxonomía que se definirá en el punto 1.4.6.

### 2.4.7 Modelo teórico software específico

Existen innumerables modelos software propuestos, prácticamente cada una de las experiencias que han sido mostradas en la sección 2.3.6 propone un modelo. Sin embargo casi todos ellos distan bastante de ser teóricos, más al contrario son implementaciones discretas para soluciones concretas. Incluso en los numerosos aportes que Danny Weyns propone en (Weyns, 2010; Weyns et al., 2009; Weyns and Holvoet, 2004; Weyns and Michel, 2015a; Weyns et al., 2007a, 2005, 2006) el modelo siempre es dependiente de la implementación. En cualquier caso las publicaciones de este investigador son un excelente punto de partida, así como las de (Dorigo et al., 2006; Dorigo and Gambardella, 1997) y (Bandini et al., 2002) y han sido una gran base sobre la que construir mi propuesta tal y como se definirá en el punto 1.4.7.

### 2.4.8 Metodología específica

Existen algunos intentos de definir una metodología aplicable a la **V.R** como en (Tanriverdi and Jacob, 2001) o (Krummel, 1998). Sin embargo todas adolecen de ser parciales, al estar centradas en un campo completo, e inmaduras. En concreto es curioso como se han intentado definir ciertas metodologías asociadas a los usos médicos de la **V.R** quizá por el gran interés en este campo, si bien es cierto que el enfoque es casi en su totalidad hardware.

Es, por tanto, necesaria una metodología amplia y «multiuso», enfocada desde una perspectiva teórica y reforzada por la práctica. Para cubrir este vacío propongo la «Metodología **PIIVE-DM**» tal y como se describirá en el punto 1.4.8.

### 2.4.9 Definición de servicios necesarios

Cuando hablamos de «servicios necesarios» nos estamos refiriendo a todos aquellos desarrollos informáticos que se habrán de desarrollar para implementar un entorno **V.I.E** maduro y funcional. Estos servicios serán, obviamente, muy dependientes de la arquitectura y la metodología utilizados en sus análisis, pero siempre se mantendrán ciertas constantes definidas por el modelo teórico.

En cualquier caso no existe dicha definición teórica, por lo que mi propuesta en este sentido, que será expuesta en el punto 1.4.9, apuesta por los **Sistema Multi Agente Situado** en **V.I.E's** como estrategia para abordar estos desarrollos software.

### 2.4.10 Plataformas de servicios

Existen numerosas plataformas de servicios **V.R**, unas públicas y abiertas y otras específicas para soluciones de nicho. No entra dentro del alcance de esta tesis proponer otra plataforma de servicios, más al contrario, en la experimentación se verá que se ha utilizado la plataforma **Open Simulator** aunque podríamos haber utilizado cualquier otra. En realidad esta necesidad está orientada a disponer de unos requisitos concretos a las plataformas de servicios existentes. También sería necesario definir cánones de compatibilidad inter-plataformas en pro de una futura universalización de los servicios.

Mi propuesta en este sentido consiste en definir estos patrones, así como diferenciar entre los distintos tipos de plataformas (de servicio o de entorno) que deberían existir, y será detallada en el punto 1.4.10.

### 2.4.11 Frameworks de servicios

Denominamos «framework de servicios» a aquel software que en forma de librería o de entorno de trabajo facilita y da un contexto a los desarrollos que se realizan en un proyecto dado. Existen numerosas propuestas tanto dentro del mundo de los sistemas multiagente como en el software genérico y de video-juegos.

En este caso mi propuesta es una implementación real software que se ha utilizado en la mayor parte de las experimentaciones documentadas en la presente tesis y que ha sido diseñada siguiendo los preceptos teóricos del modelo **DRM** y de la metodología **PIIVE-DM**, que será definida en 1.4.11.

---

# 3

## Modelo Teórico y Metodología

*«Revoloteaba alegremente; era una mariposa muy contenta de serlo. No sabía que era Chuang Tse. De repente despierta. Era Chuang Tse y se asombró de serlo. Ya no le era posible saber si era Chuang Tse que soñaba ser una mariposa, o era una mariposa que soñaba ser Chuang Tse».*

- Chuang Tse, año 300 a.C.

### 3.1 Introducción

En este capítulo vamos a incluir todas las propuestas de esta Tesis. Se propondrán metáforas y principios de diseño que creen espacios contextuales bien identificados para acabar con la terminología imprecisa, se plantearán modelos matemáticos abstractos para soportar taxonomías y arquitecturas específicas, se mostrarán metodologías experimentales para definir buenas prácticas promoviendo su uso en la comunidad software y se aportarán entornos de trabajo basados en inteligencia artificial idóneos para implementar soluciones.

En la siguiente lista se resumen de manera esquemática las propuestas basadas en las hipótesis una vez han sido caracterizadas con mi propuesta específica. De esta manera se podrá referenciar de manera más sencilla cada propuesta según su codificación GNx.

- **G1** - met. ant.: Metáfora del Entorno
- **G2a** - pr. dis.: Principio de Máxima Verosimilitud
- **G2b** - pr. dis.: Validez Ecológica

- **G2c** - pr. dis.: Flujo de Compromiso de Presencia
- **G3** - for. abs.: Entorno  $E=(S,O,F)$
- **G4** - mfm: Índice de Inmersión
- **G5** - mfm: Índice de Validez Ecológica
- **G6** - tip: Taxonomía de Servicios
- **G7a** - mts: Modelo DEC2RPM
- **G7a** - mts: Modelo **DRM**
- **G8** - metodología **PIIVE-DM**
- **G9** - def. serv.: Paradigma **S.M.A.S**
- **G10** - plat. serv: Met Práctica, Serv. Internos/Externos
- **G11a** - framework: **MetabotLib**
- **G11b** - framework: **SSR-Hub**

## 3.2 Metáfora del Entorno

Siguiendo la estrategia argumentada en el capítulo anterior vamos a utilizar la *metáfora del entorno* como línea argumental de toda esta propuesta.

**Metáfora** Figura retórica de pensamiento por medio de la cual una realidad o concepto se expresan por medio de una realidad o concepto diferentes con los que lo representado guarda cierta relación de semejanza.

Basándonos en la definición de metáfora, podemos extenderla a la ***metáfora del entorno*** como: ***buscar la máxima relación de semejanza entre un entorno original y un entorno virtual.***

A lo largo de la presente tesis volveremos en numerosas ocasiones a esta metáfora, por ejemplo a la hora de contextualizar la validez ecológica, definir el principio de máxima verosimilitud o los servicios de percepción.

### 3.2.1 Tecnología de soporte

La *metáfora del entorno* requiere de la existencia de una tecnología de soporte que permita su realización. La tecnología hardware (gafas de realidad virtual, guantes, etc.) están fuera del alcance de la presente tesis y únicamente nos referiremos a ellas en términos de madurez de producto.

Sin embargo es necesario interpretar el concepto de Tecnología más ampliamente y no circunscribirlo únicamente al hardware: la tecnología y metodología de programación que propongamos también debe ser acorde con nuestra estrategia. Por lo tanto debemos tener en cuenta las características que estamos buscando en el paradigma de programación, siempre siguiendo la *metáfora del entorno*:

1. **Concurrencia:** en los ***Entornos Virtuales Inmersivos*** es simplemente inviable que los elementos software deban ser secuenciales ya que se contravendría la metáfora, en un entorno los objetos no se comportan secuencialmente.
2. **Independencia:** los elementos software pueden ser independientes unos de otros. Aunque en ocasiones los objetos de un entorno están ligados, esto debe estar causado por un agente externo.

3. **Interactividad:** los elementos software pueden (y de hecho deben) interactuar entre ellos para generar sinergias. Además la falta de interactividad daría al traste con la metáfora.
4. **Voluntad:** en los entornos hay elementos inanimados (objetos) y elementos animados, o vivos. La característica última que diferencia unos de otros (aparte de consideraciones filosóficas y espirituales) es la existencia de voluntad o intencionalidad propia. Si bien es cierto que existen entornos en los cuales no hay elementos vivos la naturaleza social del ser humano requiere de esta característica para poder cumplir con la metáfora.

Estos cuatro requisitos de la metáfora del entorno se revisarán y ampliarán a lo largo de la presente tesis.

### 3.2.2 Principios de Diseño

#### 3.2.2.1 Validez Ecológica

La *validez ecológica* es un concepto genérico referido a la realización de pruebas, clasificada como un subtipo de validez externa (Valle, 1985). Este concepto se ha usado ampliamente en investigación médica, clínica y psicológica (Bombín-González et al., 2014) casi siempre asociada al concepto de percepción como en (Rizzo et al., 2006; Tarr and Warren, 2002) y desempeñando el papel de métrica de validez de una actividad cuando dicha actividad no se realiza en su entorno habitual. En los estudios médicos se suele argumentar que una prueba, tratamiento, entrenamiento o diagnóstico tiene una validez ecológica alta siempre que el resultado sea extrapolable de un escenario a otro. Por ejemplo, cuando un cirujano practica una intervención sobre un modelo de plástico se dice que tiene una *validez ecológica* baja, ya que existe una gran diferencia entre el entrenamiento y la operación real.

En los últimos años este concepto se ha usado ampliamente aplicado a las experiencias médicas con *Realidad Virtual* (Parsons, 2015; Tarnanas et al., 2013). Esta «migración» natural de conceptos desde un campo de estudio es absolutamente lógico ya que la **V.R** es una réplica de la realidad donde se pueden realizar sin riesgo distintas actividades médicas, como cualquier laboratorio de entrenamiento.



El concepto de **validez ecológica** va a ser revisado y citado en numerosas ocasiones a lo largo de la presente tesis, ya que se va a convertir en uno de los factores claves para medir el éxito de un **V.I.E.** En el punto 3.3.6 se definirá el *Índice de Validez Ecológica*  $I_{eco}$  y en el punto 3.4.9 se definirán los *Servicios Ecológicos*  $\Psi_{eco}$  en función de las características funcionales que un **V.I.E** debe tener para ser considerado suficientemente ecológico.

### 3.2.2.2 Principio de Máxima Verosimilitud

Este concepto toma su nombre de la herramienta estadística *Estimación por Máxima Verosimilitud* ya que el espíritu del concepto es similar. En estadística, cuando se desea ajustar un modelo y estimar sus parámetros se busca un *estimador* que sea lo más parecido posible, al que se denomina *estimador de máxima verosimilitud*.

De forma similar, en momento de diseñar un **V.I.E** nos encontramos numerosas veces con la necesidad de implementar un objeto, servicio o cualquier otro elemento desde cero. Si no se dispone de un principio de diseño claro se corre el riesgo de diseñar este elemento sin seguir un camino concreto.

Siguiendo la idea del *estimador de máxima verosimilitud* la propuesta es buscar un «estimador» dentro del mundo real que sirva como modelo para la creación a la que nos estamos enfrentando. Este «estimador» debe cumplir un servicio similar al servicio que estamos diseñando. No hay que confundir este **principio de máxima verosimilitud** con la anteriormente definida **validez ecológica**: el primero es un principio de diseño que de ser aplicado correctamente consigue que el **V.I.E** a diseñar tenga el segundo.

Una vez sentadas las bases de este principio, se define el **Principio de Máxima Verosimilitud** como: *siempre que se haya de diseñar un nuevo elemento en un V.I.E se debe buscar un elemento equivalente en la realidad y utilizarlo como referencia.*

Por ejemplo: *si en un V.I.E* debemos construir un elemento que llame la atención ante un evento dado, el principio de máxima verosimilitud nos invita a utilizar como «estimador» una sirena, una alarma, un gong o una luz destellante. El resultado de esta estrategia es que habitualmente cualquier persona sabrá interpretar el mensaje no verbal que está lanzando cualquiera de esos elementos.

Si por el contrario no seguimos este principio podríamos diseñar un «toroide rosa flotante» para indicar que ese evento ha llegado. Funcionalmente son exactamente iguales

y sin duda el toroide es más sencillo de implementar que cualquiera de los otros elementos, pero desde el punto de vista del potencial inmersivo es muy inferior, al obligar a los usuarios a aprender el significado de ese símbolo que no responde a ningún background cultural.

### 3.2.2.3 Flujo de Compromiso de Presencia

El concepto de «deep play» no tiene una buena traducción en castellano salvo puede que «compromiso con el juego», pero está asociado a la manera en la que los seres humanos nos implicamos en una actividad lúdica, independientemente del tipo de juego que sea.

Este concepto no es en absoluto nuevo, ya en 1840 Jeremy Bentham (Bentham and Dumont, 1840) se refería a él como un *estado mental* en el que los jugadores participan en el juego incluso sin un beneficio económico implícito: jugar por el mero placer de jugar.

En una interpretación del término más reciente, Diane Carr lo caracteriza en (Carr, 2006) haciendo un paralelismo en el conocido juego de rol Dragones y Mazmorras (Dungeons&Dragons): los jugadores que D&D que muestran una actitud «deep play» son capaces de aprenderse los monstruos existentes, los diferentes tipos de magia o las características de cada armadura, mientras que los jugadores con una actitud más despreocupada por el contexto del juego suelen utilizar más la técnica de «abrirse camino a golpes».

Este «deep play» también llega a darse en actividades no interactivas como libros, series de televisión o películas: aprender Klingon, conocer la genealogía completa de la casa Stark o saber diferenciar entre un Maier y un Valar son actividades que los seres humanos llegamos a realizar voluntariamente, sin un ánimo de recompensa directa o indirecta salvo el mero y profundo placer de participar de un consenso a sabiendas irreal.

Acercando este término a la **V.R** podemos apoyarnos en él para describir el estado mental objetivo al construir un **V.I.E**. En la ciencia ficción Vernor Vinge en (Vinge, 1981) su novela «True Names» describe un interfaz inmersivo a nivel neuronal que en lugar de usar la clásica percepción ocular utilizaba un concepto similar estimulando las partes del cerebro que son capaces de tratar con las formas abstractas de los conceptos discretos, similar al que ocurre cuando al leer un libro nos sentimos inmersos en él.

Este término «deep play» también se ha estudiado desde el mundo de los videojuegos y la **Realidad Virtual**. La idea es que una vez se ha conseguido el estado mental «deep

play» se debe mantener durante todo el tiempo posible.

Una vez el usuario se introduce dentro del **Entorno Virtual Inmersivo** se establece un flujo de compromiso de presencia (McMahan, 2003; Xu et al., 2017) que ha de ser mantenido durante todo el tiempo de la experiencia. Dicho flujo consiste en que toda la información e interacción del usuario con el entorno virtual debe permanecer constantemente alimentando la ilusión inmersiva, evitando cualquier interrupción.

Una vez sentadas las bases de este principio, se define el **Flujo de Compromiso de Presencia** como: *un principio de diseño para V.I.E's que consiste en dos pasos: (1) dotar al V.I.E de la máxima cantidad de información de «background» posible para facilitar al usuario llegar al estado mental de «deep play» o «compromiso de presencia» y (2) mantener un flujo constante y coherente con esa información de «background» evitando en lo posible interrupciones bruscas o información incoherente .*

El objetivo general de este principio es aumentar la **capacidad inmersiva** de un **V.I.E**

### 3.3 Formalización abstracta del modelo

Una vez hemos sentado las bases del problema al que nos enfrentamos, hemos definido una estrategia y establecido las herramientas tecnológicas que vamos a utilizar, es el momento de formalizar los elementos clave para poder enunciar un modelo coherente.

#### 3.3.1 Descripción formal de entorno

El primer concepto que vamos a modelizar es el de entorno para que nos sirva como cimientos para el resto de argumentación. Si buscamos referencias al concepto de entorno dentro de las matemáticas (Bourbaki, 1998) (Runde, 2006) nos encontramos que la idea de entorno (o vecindad) está íntimamente ligado al *espacio topológico*. En este contexto se define entorno como un conjunto de posiciones accesibles entre ellas, es decir, aquel espacio que exhibe continuidad y que permite la existencia de al menos un posible camino entre dos puntos A y B cualesquiera.

En (Wooldridge, 2002) se propone una arquitectura abstracta para la modelización de agentes inteligentes en la que se define el **entorno** como un conjunto finito de estados instantáneos y discretos:  $E = e, e', \dots$ . Sin embargo esta definición es demasiado simple y no es capaz de soportar la definición abstracta de **entorno** necesaria para el problema que nos ocupa.

Definimos por tanto aquí una abstracción de *entorno*  $\mathbb{E}$  como una terna de estados  $\mathbb{S}$ , objetos  $\Omega$  y servicios  $\Psi$ :

$$\mathbb{E} = \{\mathbb{S}, \Omega, \Psi\}$$

##### 3.3.1.1 Estados de un Entorno

Mediante el concepto de *estado* vamos a modelizar la naturaleza discreta y secuencial de un *entorno*. El estado representa un instante concreto en la secuencia de sucesos de un entorno. Desde una interpretación más cercana a la física clásica podríamos asimilar el concepto de *Estado* con el de *Tiempo*. Sin embargo no sería correcto hacerlo directamente ya que el objetivo de toda este modelo abstracto persigue definir cómo los sistemas computacionales implementan aplicaciones de tipo «simulador», no definir una física alternativa.

Los otros dos componentes de la terna (objetos y servicios) siempre están ligados a un estado concreto, es decir, no podremos decir que un objeto o un servicio tienen un valor discreto salvo que lo referenciamos en un estado. Dado que los estados tienen valores concretos, secuenciales y ordenados, podremos definir el concepto de precedencia entre estados, fundamental para cumplir el principio de causalidad.

Sea  $\mathbb{S}$  un conjunto finito indeterminado de estados  $s$

$$\mathbb{S} = \{s_1, s_2, \dots, s_n\} \quad / \quad s_i \in \mathbb{Z} \quad / \quad \mathbb{S} \subset \mathbb{Z}$$

que satisfacen de dos en dos la condición de secuencialidad en el dominio de los números enteros

$$\{s_1, s_2 \in \mathbb{S} : s_1, s_2 \in \mathbb{Z} \wedge s_1 \leq s_2\}$$

por lo tanto podemos ver  $\mathbb{S}$  como un vector de enteros creciente

### 3.3.1.2 Objetos de un Entorno

Un entorno tiene un carácter de contenedor implícito. La idea de *objeto* responde al concepto de «aquello que está contenido en un entorno» de una manera abstracta. Desde un punto de vista computacional podemos asimilar los objetos a los datos de un sistema, el contenido estático sobre los que operan los procesos.

Sea el conjunto de todos los objetos de un entorno, notado como  $\Omega$ , formado por una serie finita de objetos  $\omega$

$$\omega = \{\omega_1, \omega_2, \dots, \omega_n\} \quad / \quad \omega_i \in \Omega$$

Se define a su vez un objeto  $\omega$  como una dupla de propiedades  $\{P, Q\}$ , siendo  $P$  las propiedades posicionales y  $Q$  las propiedades no posicionales del objeto.

$$\omega = \{P, Q\}$$

Atendiendo a esta diferenciación  $P_\Omega$  será un vector de valores del dominio de los números  $\mathbb{R}$ . Mediante este vector se representan las propiedades posicionales de los

objetos y se define como

$$P = \{p_1, p_2, \dots, p_n\} \quad / \quad p_i \in \mathbb{R}$$

Definiremos también  $Q$  como un conjunto de características no posicionales y se define como

$$Q = \{q_1, q_2, \dots, q_n\}$$

**Dimensionalidad de un objeto** Podremos hablar sobre la dimensionalidad de un objeto en función de la ordinalidad del conjunto  $P$ , denotándolo con la función  $\sigma$

$$\sigma(\omega) = |P|$$

A su vez podremos hablar de entorno homodimensionales cuando todos los objetos de un mismo entorno tengan la misma dimensión, o por el contrario, de entornos heterodimensionales cuando no se de esta condición. Sea por tanto un entorno cualquiera  $\mathbb{E}$  homodimensional cuando

$$\forall \omega_i, \omega_j \in \Omega_{\mathbb{E}} : \sigma(\omega_i) = \sigma(\omega_j)$$

**Distancias entre Objetos** Dentro de nuestra propuesta podremos distinguir entre dos tipos de distancia: la distancia espacial y la distancia cualitativa. El primer tipo, la espacial, se aplicará sobre el conjunto de propiedades posicionales  $P$  del objeto y se podrá calcular con cualquier método de medida de distancia apropiada al tipo de entorno. Por ejemplo, en los entornos que describan un espacio euclídeo (de cualquier dimensión) ver punto 3.3.3.1 podremos utilizar la distancia euclídea:

$$d_E(\omega, \omega') = \sqrt{\sum_{i=1}^{\sigma(\omega)} (\omega_i - \omega'_i)^2}$$

En cuanto a la distancia cualitativa, la utilizaremos para caracterizar el concepto de similitud entre objetos a través de la comparación de sus características. Se define por tanto el concepto de *distancia cualitativa* entre dos objetos a través del método conocido

como «Distancia de Mahalanobis» (Portillo and Mendoza, 2015) usado ampliamente para medir la similitud entre variables aleatorias multidimensionales:

$$d_m(\vec{x} - \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T \Sigma^{-1} (\vec{x} - \vec{y})}$$

Si asimilamos el conjunto de propiedades no posicionales  $Q$  a un vector de valores  $\vec{x}$  podremos, por isomorfismo, podemos definir la distancia cualitativa entre dos objetos  $\omega_a$  y  $\omega_b$  como:

$$\|\omega_a - \omega_b\| = \sqrt{(\vec{Q}_a - \vec{Q}_b)^T \Sigma^{-1} (\vec{Q}_a - \vec{Q}_b)}$$

siendo  $\Sigma^{-1}$  la inversa de la matriz de covarianza de  $\vec{Q}_a$  y  $\vec{Q}_b$ .

### 3.3.1.3 Servicios de un Entorno

Por último vamos a definir la abstracción de servicio de un entorno como, *todas aquellas acciones transformadoras que permiten a los objetos evolucionar a lo largo de los estados circunscritos a un entorno dado*. El principio de causalidad, clave para la comprensión intuitiva antropomórfica de un entorno, se expresa a través de este concepto.

Pero siguiendo la metáfora del entorno, vamos a trabajar con otra definición más acorde al nuevo paradigma y de la que se extrae su nombre: *todos los servicios que un entorno ofrece a los elementos que lo habitan*. El conjunto de servicios de un entorno será notado como  $\Psi$ .

Vamos a necesitar una extensión de este concepto para poder definir su alcance aunque entraremos a detallarlo en el siguiente punto. Sea

$$\Psi^* = \{\psi \in \Psi^* : \psi \text{ computable}\}$$

Definimos entonces los servicios de un entorno dado como

$$\Psi = \{\psi_1, \psi_2, \dots, \psi_n\} \quad / \quad \psi_i \in \Psi^* \quad / \quad \Psi \subseteq \Psi^*$$

Donde los servicios  $\psi$  se definen como un sistema compuesto de dos cuerpos excluyentes entre sí. O bien un servicio se expresa como una función con dos parámetros

denominándose «servicio final», o bien se expresa como un agregado de servicios subordinados  $\psi'$  denominándose «servicio abstracto».

En el caso de servicios finales  $\psi$  tendrá siempre dos parámetros: un objeto  $\omega$  y un estado inicial  $s_i$  obteniendo un objeto  $\omega'$  y un nuevo estado final  $s_{i+1}$ :

$$\psi(\omega, s_i) \Rightarrow \begin{cases} \psi'_1(\omega, s_i) \oplus \psi'_2(\omega, s_i) \oplus \dots \oplus \psi'_n(\omega, s_i) & \text{si } \psi \text{ abstracto} \\ \{\omega', s_{i+1}\} : \{\omega, \omega' \in \Omega; s_i, s_{i+1} \in \mathbb{S}\} & \text{si } \psi \text{ final} \end{cases}$$

Cumpléndose que la diferencia entre el objeto  $\omega'$  y  $\omega$  sea *Parte* de la unión de ambos objetos. Es decir, el contenido de ambos objetos puede ser exactamente igual, completamente distinto o parte igual y parte distinta:

$$\omega' - \omega = \mathcal{P}(\omega \cup \omega')$$

Cabe aclarar que las funciones  $\psi$  pueden ser aplicadas a cualesquiera objeto  $\omega$  en el mismo estado  $s_i$  de manera concurrente

$$\forall s \in \mathbb{S}; \forall \omega_i \in \Omega; \forall \psi_j \in \Psi / \exists \psi_j(s, \omega_i)$$

Para abstraer un método de comparación entre servicios, definimos una función de distancia entre dos servicios dados  $\psi$  y  $\psi'$  que notaremos como  $\|\psi_a - \psi_b\|$ . En el caso de servicios  *finales*, es decir, que no están formados por otros subservicios, la función de comparación se define como la *distancia cualitativa* entre los dos objetos resultantes de aplicar el servicio. En el caso contrario, los servicios abstractos, se calculará como la suma de las distancias de los subservicios.

$$\|\psi_a - \psi_b\| = \begin{cases} \sum_{\forall x \in \psi_a; \forall y \in \psi_b} \|x - y\| & \text{si } \psi = \{\psi'_1, \psi'_2, \dots, \psi'_n\} \\ \|\omega'_a - \omega'_b\| & \text{si } \psi \neq \{\psi'_1, \psi'_2, \dots, \psi'_n\} \end{cases}$$

### 3.3.2 Medidas de similitud entre Servicios

Ya se ha definido en el punto 3.3.1.3 la función de distancia entre servicios. Teniendo en cuenta la necesidad de poder comparar servicios de una manera eficiente se formularán a continuación dos métricas: la *función de similitud simple* ( $s^\psi$ ) y la *función de similitud inyectiva* ( $w^{A \rightarrow B}$ ).



### 3.3.2.1 Función de similitud simple

Tomando como base la distancia entre servicios ( $\|\psi_a - \psi_b\|$ ), es posible definir el concepto de similitud mediante la función  $s^\psi$  que aplica a dos servicios  $\psi_a$  y  $\psi_b$  cualesquiera y cuyo valor de salida está normalizado entre 0 y 1. Un valor de 1 o muy cercano a 1 significará que ambos servicios son iguales, mientras que un valor de 0 o cercano a 0 significará que ambos servicios son diferentes. Se formula tal que:

$$s^\psi(\psi_a, \psi_b) = \left( \frac{1}{\|\psi_a - \psi_b\| + 1} \right)^\mu$$

siendo  $\mu$  el *umbral de proximidad*, un valor empírico que ha de ser ajustado en función de *entorno* que da contexto a los servicios.

### 3.3.2.2 Condiciones de similitud entre conjuntos de servicios

Cuando queremos comparar conjuntos de servicios, ya sea refiriéndose a  $\Psi$  según la definición básica de entorno tal y como se define en 3.3.1  $\mathbb{E} = \{\mathbb{S}, \Omega, \Psi\}$ , o sea refiriéndose a un subconjunto cualquiera de  $\Psi$  con al menos un servicio. Se deberán cumplir las dos condiciones siguientes:

- *Condición 1:* Deberá existir una función que relacione los conjuntos de servicios A y B y responda a la pregunta «¿a qué elemento de B le corresponde este elemento de A?» tal que

$$f : A \rightarrow B$$

- *Condición 2:* Dicha función debe ser sobreyectiva tal que

$$\forall a \in A, \exists b \in B : f(a) = b$$

Siguiendo la notación de servicio que hemos propuesto,

$$f_{\Psi}^{A \rightarrow B} : \Psi^A \rightarrow \Psi^B$$

$$\forall \psi^a \in \Psi^A, \exists \psi^b \in \Psi^B : f_{\Psi}^{A \rightarrow B}(\psi^a) = \psi^b$$

### 3.3.2.3 Función de similitud sobreyectiva

La función de similitud sobreyectiva es una personalización de la simplificada definida en 3.3.2.1. En este caso, en lugar de aplicarse entre dos servicios se aplica entre un servicio y un entorno. La notaremos como  $w^{A \rightarrow B}(\psi^A, \Psi^B)$

Apoyándonos en las condiciones definidas en 3.3.2.2 contamos con la función de relación sobreyectiva  $f_{\Psi}^{A \rightarrow B}$  que nos asegura que para  $\psi^A$  existe  $\dot{\psi}^B \in \Psi^B$  con el que está relacionado.

Mediante dicha función se identificará el servicio asociado que se identificará como  $\dot{\psi}^B$  tal que

$$\dot{\psi}^B = f_{\Psi}^{A \rightarrow B}(\psi^A)$$

Sea por lo tanto la función de similitud sobreyectiva de servicios ( $w^{A \rightarrow B}(\psi^A, \Psi^B)$ ) definida como

$$w^{A \rightarrow B}(\psi^A, \Psi^B) = s^{\psi}(\psi^A, \dot{\psi}^B) = s^{\psi}(\psi^A, f_{\Psi}^{A \rightarrow B}(\psi^A))$$

Desarrollando la *función de similitud simple* según se ha definido en 3.3.2.1 ( $s^{\psi}$ )

$$w^{A \rightarrow B}(\psi^A, \Psi^B) = \frac{1}{\|\psi^A - f_{\Psi}^{A \rightarrow B}(\psi^A)\| + 1}$$

quedando esta función como indicador numérico acotado entre (0, 1) de la semejanza entre un servicio  $\psi^A \in A$  y su servicio asociado en  $B$ .

### 3.3.2.4 Función de similitud entre entornos

Por último, dentro de las medidas de similitud, vamos a definir la función de similitud entre entornos basándonos en las anteriores funciones.

Partiendo de la premisa en la que se cumplen las *condiciones de similitud* indicadas en el punto 3.3.2.2 vamos a definir la *función de similitud entre entornos* ( $\Delta^{A \rightarrow B}(\mathbb{E}^A, \mathbb{E}^B)$ ) como

$$\Delta^{A \rightarrow B}(\mathbb{E}^A, \mathbb{E}^B) = \sum_{i=1}^{|\Psi^A|} w^{A \rightarrow B}(\psi_i^A, \Psi^B) \quad / \quad \forall \psi^A \in \Psi^A \in \mathbb{E}^A, \Psi^B \in \mathbb{E}^B$$

Esta función calcula la suma de todas las funciones de similitud sobreyectiva entre los servicios del entorno A y el entorno B. Al obligar a que exista una función sobreyectiva de  $A \rightarrow B$  aseguramos que dicha función exista y pueda devolver un valor, aunque este sea 0.

El resultado será por tanto una suma de  $|\Psi^A|$  elementos con valores en el rango  $(0, 1)$ , es decir, el rango final de la función será  $(0, |\Psi^A|) \in \mathbb{R}^+$ .

Si desarrollamos esta formulación completamente podríamos expresar la función de similitud entre entornos como:

$$\Delta^{A \rightarrow B}(\mathbb{E}^A, \mathbb{E}^B) = \sum_{i=1}^{|\Psi^A|} \frac{1}{\|\psi_i^A - f_{\Psi}^{A \rightarrow B}(\psi_i^A)\| + 1}$$

$$\forall \psi^A \in \Psi^A \in \mathbb{E}^A, \Psi^B \in \mathbb{E}^B$$

### 3.3.3 Casos particulares de entorno

Llegados a este punto es necesario recordar que toda esta modelización abstracta va dirigida únicamente a definir los conceptos de entorno, inmersión y validez ecológica. Estos conceptos son siempre interpretados desde un punto de vista antropomórfico y sujetos por tanto a la percepción humana. No debe interpretarse este trabajo como un intento de formalizar una física alternativa para entornos reales o virtuales, sino como un experimento teórico que nos ayude a comprender de la manera más abstracta posible cómo definir los **Entornos Virtuales Inmersivos** y la **Realidad Virtual**.

Siguiendo con la premisa de la metáfora del entorno, debemos observar algunos tipos de entorno muy particulares e importantes para nuestro estudio. Obtengamos ahora la abstracción de todos los entornos posibles, notándola como  $\mathbb{E}^*$  y definiéndola tal que

$$\mathbb{E}^* = \{\mathbb{E}_1, \mathbb{E}_2, \dots, \mathbb{E}_n\}$$

Dentro todos los entornos posibles en  $\mathbb{E}^*$  existe un subconjunto formado por todos los *entornos* que cumplen la condición de ser entornos euclídeos de grado 3, usualmente denominados como  $\mathbb{R}^3$  aunque por homogeneizar la nomenclatura nos referiremos a este subconjunto también como  $\mathbb{E}^3$ .

$$\mathbb{E}^3 = \{\mathbb{E} \in \mathbb{E}^* : \mathbb{E} \text{ es entorno euclídeo grado } 3\}$$

### 3.3.3.1 Entornos euclídeos de grado 3

Para poder evaluar la condición «ser entorno euclídeo de grado 3» vamos a tomar la notación usada habitualmente en álgebra topológica para referirse a estos entornos. Si  $\mathbb{R}^n$  es un entorno euclídeo de grado  $n$ ,  $\mathbb{R}^3$  será como notemos a los de grado tres.

$$\mathbb{R}^3 = \mathbb{E}^3 \subseteq \mathbb{E}^*$$

Queda probado por tanto que los entornos euclídeos de grado 3 son un subconjunto de todos los posibles entornos cualesquiera su grado. Siguiendo la generalización de los espacios euclídeos se definen como un vector  $n$ -dimensional

$$\mathbb{R}^n = \{x_1, x_2, \dots, x_n\} / x_i \in \mathbb{R}$$

lo que permite generalizar principios geométricos básicos como la función de la distancia

$$d(x, y) = \|x - y\| = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

o el producto escalar

$$x \cdot y = \sum_{i=1}^n x_i y_i = x_1 y_1 + x_2 y_2 + \dots x_n y_n$$

Pero volviendo a la definición de condición «ser entorno euclídeo de grado 3», si concretamos la definición vectorial general de grado 3 tendremos que

$$\mathbb{R}^3 = \{x_1, x_2, x_3\} / x_1, x_2, x_3 \in \mathbb{R}$$

que si comparamos con la definición anterior de propiedades físicas  $P$  de los objetos  $\Omega$  de un entorno  $\mathbb{E}$

$$P = \{\rho_1, \rho_2, \dots, \rho_n\} \quad / \quad \rho_i \in \mathbb{R}$$

y acotamos a aquellas propiedades que formen vectores de tres elementos

$$P^3 = \{\rho_1, \rho_2, \rho_3\} \quad / \quad \rho_i \in \mathbb{R}$$

es notable que las propiedades de un entorno formadas por un vector de tres elementos y la definición vectorial de un entorno euclídeo de grado tres comparten estructura isomórfica:

$$\mathbb{E}^3 \Rightarrow \mathbb{R}^3 \quad / \quad h(\rho_1, \rho_2, \rho_3) \Rightarrow h'(x_1, x_2, x_3)$$

sin embargo para poder establecer una función isomórfica biyectiva  $\mathbb{E}^3$  también debe poder ser considerado un entorno euclídeo, para lo cual deben cumplirse los cinco postulados de Euclides (Beckmann, 2015) y curvatura cero.

Los cinco postulados son:

1. Dos puntos cualesquiera determinan un segmento de recta.
2. Un segmento de recta se puede extender indefinidamente en una línea recta.
3. Se puede trazar una circunferencia dados un centro y un radio cualquiera.
4. Todos los ángulos rectos son iguales entre sí.
5. Postulado de las paralelas. Si una línea recta corta a otras dos, de tal manera que la suma de los dos ángulos interiores del mismo lado sea menor que dos rectos, las otras dos rectas se cortan, al prolongarlas, por el lado en el que están los ángulos menores que dos rectos.

Adicionalmente la condición de «curvatura cero» se define como: *la suma de los tres ángulos interiores de un triángulo es siempre 180 grados.*

Para facilitar su formalización definiremos la función *espacioEuclideo*(A) que aplicado sobre un espacio dado informa sobre la satisfacción de los cinco postulados de Euclides y la condición de curvatura cero.

Por lo tanto, y siguiendo la premisa de la metáfora del entorno en la que los seres humanos no estamos capacitados para la experimentación directa de entornos euclidianos de grado distinto de tres, definimos un caso particular de entorno sobre el que nos vamos a centrar durante el resto de la tesis, notándolo como  $\mathbb{E}^3$  y siendo matemáticamente definible como

$$\mathbb{E}^3 = \{\mathbb{E} \in \mathbb{E}^* : \{P \in \mathbb{E} : |P| = 3\} \wedge \text{espacioEuclideo}(\mathbb{E})\}$$

Sea por tanto  $\mathbb{E}^3$  el conjunto de todos los posibles espacios euclidianos de grado tres, también referidos comúnmente como **entornos tridimensionales**

### 3.3.3.2 Entorno real

Una vez más debemos acudir a la metáfora del entorno para buscar la formalización de un entorno particular que vamos a necesitar para sentar las bases matemáticas del modelo: el entorno real.

Ya hemos definido que la abstracción de entorno que estamos proponiendo cumple que  $\mathbb{E}^3$  es un entorno euclidiano equivalente al concepto antropomórfico de realidad. Aunque en este punto podríamos argumentar sobre el concepto metafísico de realidad o sobre la naturaleza física infinitamente más compleja de la realidad, eso se escapa del alcance de esta tesis. Nos centraremos en la percepción antropomórfica de la realidad siguiendo nuestra estrategia inicial.

Por lo tanto podemos postular que dentro del conjunto de todos los entornos que son entornos euclídeos de grado 3 ( $\mathbb{E}^3$ ) existe un entorno especial: el mundo real. Basándonos en el principio natural antropomórfico que define el universo en el que los seres humanos nos desenvolvemos y del que somos parte como un entorno tridimensional euclídeo y newtoniano, definimos un entorno especial que denominaremos *entorno real* y notaremos  $\mathbb{E}^{\mathfrak{R}}$ .

El *entorno real* es un subconjunto impropio de los entornos euclídeos de grado 3, siendo la cardinalidad de este *entorno real* siempre igual a 1:

$$\mathbb{E}^{\mathfrak{R}} \subset \mathbb{E}^3 / |\mathbb{E}^{\mathfrak{R}}| = 1$$

Mediante el principio del tercero excluido definiremos otro subconjunto de entornos como complementario de  $\mathbb{E}^{\mathfrak{R}}$  en  $\mathbb{E}^3$ , el subconjunto de *entornos virtuales*

$$\begin{aligned}\mathbb{E}^3 &= \mathbb{E}^{\mathfrak{R}} \cup \left(\mathbb{E}^{\mathfrak{R}}\right)^c \\ \left(\mathbb{E}^{\mathfrak{R}}\right)^c &= \mathbb{E}^3 \setminus \mathbb{E}^{\mathfrak{R}} \\ \left(\mathbb{E}^{\mathfrak{R}}\right)^c &= \mathbb{E}^{\mathfrak{V}} \\ \mathbb{E}^3 &= \mathbb{E}^{\mathfrak{R}} \cup \mathbb{E}^{\mathfrak{V}}\end{aligned}$$

El uso del vocablo «virtual» se toma de manera natural según su definición «*Que solamente existe de forma aparente y no es real*»

### 3.3.4 Índice de Inmersión

Definimos *índice de inmersión* como un concepto basado en las *medidas de similitud entre servicios* tal y como se han definido en el punto 3.3.2 y aplicado a cuantificar el potencial de inmersión de un *entorno*. Aunque el concepto de inmersión se define exhaustivamente en el punto 3.3.5 más adelante, baste en este punto aclarar que denominaremos  $\Psi_\Phi$  a un subconjunto de servicios de un entorno que engloban las características de inmersión tal y como se definirá en detalle en el punto 3.4.

En el punto 3.3.2.4 se ha definido la *función de similitud entre entornos*  $(\Delta^{A \rightarrow B}(\mathbb{E}^A, \mathbb{E}^B))$  y basándonos en esta métrica podremos formular el *índice de inmersión* ( $I_\Phi$ ) de un *entorno* ( $\mathbb{E}^\nu$ ) como una función que arroja un valor numérico entre 0 y 1 indicando el nivel de inmersión que dicho *entorno* es capaz de ofrecer tal que

$$I_\Phi(\mathbb{E}^\nu) = \frac{\Delta_\Phi^{A \rightarrow B}(\mathbb{E}^{\mathfrak{R}}, \mathbb{E}^\nu)}{|\Psi_\Phi^{\mathfrak{R}}|}$$

dado que tal y como se define en el punto 3.3.2.4 el rango de salida de la función  $\Delta^{A \rightarrow B}(\mathbb{E}^A, \mathbb{E}^B)$  es  $(0, |\Psi^A|) \in \mathbb{R}^+$  que en nuestro caso es  $|\Psi_\Phi^{\mathfrak{R}}|$  podemos afirmar que

$$I_\Phi = \left( \frac{0, |\Psi_\Phi^{\mathfrak{R}}|}{|\Psi_\Phi^{\mathfrak{R}}|} \right) \Leftrightarrow \forall x \in \mathbb{R}^+ : 0 \leq x \leq 1$$

$$I_\Phi = \left( \frac{0, 1}{1} \right) \Leftrightarrow \forall x \in \mathbb{R}^+ : 0 \leq x \leq 1$$

$$I_{\Phi} = (0, 1) \Leftrightarrow \forall x \in \mathbb{R}^+ : 0 \leq x \leq 1$$

### 3.3.5 Entornos Virtuales Inmersivos

Siguiendo el camino de la modelización abstracta de los casos particulares de entorno vamos a tratar ahora con los **Entornos Virtuales Inmersivos**.

Que un entorno virtual sea inmersivo o no depende de distintos factores, unos internos al entorno y otros externos. La tecnología de soporte de la metáfora (como se ha definido en el punto 3.2 ) es un factor absolutamente determinante. Sin embargo para el caso que nos ocupa en este momento, definir un modelo abstracto, vamos a enunciar la *asunción del canal perfecto*: ***asumimos que en el momento de medir la experimentabilidad del entorno existe una tecnología de soporte que no la limita, de igual manera que se da por sentado que el sujeto experimentador no sufre ningún impedimento sensorial que pueda limitar la experimentabilidad.***

El grado de experimentabilidad de un entorno queda por tanto limitada a los servicios del entorno a estudio, en comparación con los servicios del entorno real.

Tomando como punto de partida un *entorno virtual* cualquiera dentro del conjunto de todos los entornos virtuales  $\mathbb{E}^{\mathfrak{V}}$

$$\mathbb{E}^v \in \mathbb{E}^{\mathfrak{V}} / |\mathbb{E}^v| = 1$$

vamos a definir un subconjunto que cumpla la cualidad *inmersiva* de manera abstracta. Para ello vamos a utilizar el *índice de inmersión* definido en el punto (3.3.4).

Siendo  $\Psi_{\Phi}^{\mathfrak{R}}$  el conjunto de todos los servicios de suborden perceptivo en el mundo real

$$\Psi_{\Phi}^{\mathfrak{R}} = \{ \Psi \in \Psi^{\mathfrak{R}} : \Psi \odot \Phi \}$$

y  $\Psi_{\Phi}^v$  el conjunto de todos los servicios de  $\mathbb{E}^v$  de tipo perceptivo

$$\Psi_{\Phi}^v = \{ \Psi \in \Psi^v : \Psi \odot \Phi \}$$

podemos evaluar de manera abstracta el potencial inmersivo de  $\mathbb{E}^v$  mediante el *índice de inmersión* ( $I_{\Phi}$ ) y de esta manera caracterizar la condición de inclusión de un entorno



dentro del conjunto de *Entornos Virtuales Inmersivos* tal que

$$\mathbb{E}^{\vee} = \{\mathbb{E} \in \mathbb{E}^{\vee} : I_{\Phi}(\mathbb{E}) \geq \gamma\}$$

siendo  $\gamma$  un factor de ajuste empírico en el rango  $(0, 1)$ .

Volviendo al concepto de *entorno real*, de las anteriores definiciones se desprende que el índice de inmersión del mundo real es siempre la unidad

$$I_{\Phi}(\mathbb{E}^{\Re}) = 1$$

Lo cual es ciertamente intuitivo ya que viene a decir que la capacidad de inmersión en el mundo real es completa.

### 3.3.6 Índice de Validez Ecológica

De manera análoga al *índice de inmersión* vamos a definir el *índice de validez ecológica* siguiendo el mismo método, pero aplicado a un subconjunto distinto de servicios. En lugar de circunscribirnos a los servicios perceptivos vamos a agrupar a todos los servicios ecológicos

Denominaremos  $\Psi_{eco}$  al conjunto de servicios ecológicos de un entorno tal y como se definirán en la taxonomía del punto 3.4.

$$I_{eco}(\mathbb{E}^{\vee}) = \frac{\Delta_{eco}^{A \rightarrow B}(\mathbb{E}^{\Re}, \mathbb{E}^{\vee})}{|\Psi_{eco}^{\Re}|}$$

Como es lógico, para poder calcular el *índice de validez ecológica* es necesario que se cumplan las *condiciones de similitud entre conjuntos de servicios* tal y como se expone en el punto 3.3.2.2

## 3.4 Taxonomía de Servicios

En el presente punto vamos a identificar los servicios partiendo de la base de la metáfora del entorno para posteriormente enunciarlos formalmente.

Hemos de recordar que para poder comparar dos entornos se han de cumplir ciertas condiciones y definir ciertos valores y funciones:

- *Condición 1:* Deberá existir una correspondencia entre los servicios de ambos entornos, al menos en el contexto de la comparación.
- *Condición 2:* Debemos fijar un valor empírico para el *valor*  $\mu$  tal y como se define en 3.3.2.1. Por defecto este valor será  $\mu = 1$
- *Condición 3:* Debemos también fijar un valor  $\gamma$  como un indicador empírico del punto a partir del cual se considera un entorno inmersivo tal y como se define en el punto 3.3.5

### 3.4.1 Clasificación general de los servicios

En los siguientes capítulos vamos a estudiar los posibles servicios existentes desde distintas perspectivas, pero para poder ir elaborando una taxonomía ordenada planteamos una categorización básica de todos los servicios que puedan encontrarse en un *entorno* en el sentido más amplio del término, es decir, sin centrarnos en exclusiva en el entorno real.

Se plantea por lo tanto que existen dos categorías de primer orden:

- *Ecológicos:* Todos aquellos elementos que son naturales y desde un punto de vista antropomórfico, intuitivos.
- *Tecnológicos:* Todos aquellos elementos que han sido contruidos por el hombre.

La relevancia de distinguir entre servicios ecológicos y tecnológicos a alto nivel persigue dos propósitos:

- Poder identificar los entornos que cumplen el principio de validez ecológica

- Identificar dos categorías complementarias que unidas forman un todo. En otras palabras, de todos los servicios existentes en cualquier entorno podremos decir que son *ecológicos* o bien decir que son *tecnológicos* pero nunca podremos decir que es ambas cosas al mismo tiempo ni existe otra tercera categoría.

### 3.4.2 Clasificación de Servicios Ecológicos

Es importante no confundir los términos «ecología» y «ecologismo». El primero es *La parte de la biología que estudia las relaciones de los seres vivos entre sí y con el medio en el que viven. Relación que se da entre los seres vivos de una zona determinada y el medio en el que viven.*, mientras que el «ecologismo» es un movimiento ideológico, moral y/o político.

Partiendo de la idea de reflejar cómo los seres humanos nos relacionamos con nuestro entorno vamos a definir los servicios «ecológicos». Los servicios ecológicos son los más cercanos al antropomorfismo y a la psicología humana, englobando conceptos como la propiocepción, la auto-imagen, la conciencia social, las interacciones con otros seres humanos, etc.

Con esta premisa antropomórfica nos centraremos en la observación del entorno real natural para extraer servicios de un nivel óptimo para nuestra tarea y definimos tres subtipos de los entornos ecológicos: físicos, perceptivos e interactivos

$$Ecológicos \left\{ \begin{array}{l} Físicos \\ Perceptivos \\ Interactivos \end{array} \right.$$

Como es lógico, gran parte de los servicios ecológicos que se esperan al utilizar un *entorno virtual* son los derivados de las propiedades físicas de la materia en el mundo real. Por lo tanto vamos a reflexionar sobre qué servicios físicos podemos extraer de  $\mathbb{E}^n$ , recordando que los referenciamos matemáticamente como  $\Psi^n$ .

### 3.4.3 Identificación de servicios ecológicos físicos a partir de la observación del mundo real

El entorno real  $\mathbb{E}^{\mathfrak{R}}$  es especialmente importante ya que siguiendo la *metáfora del entorno* deberemos usarlo como referente. El entorno real nos servirá como modelo del cual extraer los tipos de servicios que los seres humanos vamos a requerir en los entornos virtuales.

Para llevar a cabo este análisis del mundo real en busca de los servicios utilizaremos el razonamiento inductivo y seguiremos un modelo «bottom-up», partiendo desde el nivel más bajo hasta los servicios más conceptuales. A la pregunta ¿cómo de alto tiene que ser el nivel conceptual de los servicios que buscamos? la respuesta es: lo suficientemente alto como para ser computable en un simulador actual en tiempo real.

Supongamos un caso extremo: un simulador exacto tal y como lo definió Richard Feynman (Feynman, 1982). Dado que la realidad responde a un sistema cuántico, sería necesario un computador cuántico para poder simularlo de manera exacta. Si recordamos el Índice de Inmersión definido en 3.3.4 en este caso  $I_{\Phi}(\mathbb{E}^{\vee}) = I_{\Phi}(\mathbb{E}^{\mathfrak{R}}) = 1$  o visto de otra manera, no sería posible distinguir entre  $\mathbb{E}^{\vee}$  y  $\mathbb{E}^{\mathfrak{R}}$ . Estaríamos hablando del objetivo ideal de nuestro modelo, la inmersión perfecta. Sin embargo en la escala en la que tanto la *metáfora del entorno* como la percepción antropomórfica son válidas sería irrelevante hacer una simulación cuántica, por ejemplo de una manzana que cae por efecto de la gravedad. Nos referiremos a este caso de simulación cuántica imposible con el símbolo  $\mathbb{Q}$

Supongamos también el otro extremo: un simulador completamente inexacto. En este caso  $I_{\Phi}(\mathbb{E}^{\vee}) = 0$  y sería el peor caso posible a imaginar. Por contra también sería el más fácil de implementar ya que el simulador no requeriría cubrir ningún servicio, pero a cambio su utilidad es inexistente y no tiene lugar seguir tratando con este caso. Nos referiremos a este simulador con el símbolo  $\emptyset$

Tendríamos entonces dos simuladores teóricos que serían irrelevantes, uno por su inexactitud absoluta y otro por su exactitud completa. Podremos entonces imaginar un espacio continuo en el rango  $[0, 1]$ . En el extremo izquierdo (0) estaría  $\emptyset$  y en el extremo derecho (1) estaría  $\mathbb{Q}$

$$\left\{ \emptyset \quad \begin{array}{c} \text{<=====}> \\ \gamma \end{array} \quad \mathbb{Q} \right\}$$

Recordemos que los entornos virtuales inmersivos estaban caracterizados por devolver

un valor en la función de *Índice de Inmersión* mayor que un factor  $\gamma$  empírico ( $\mathbb{E}^V = \{\mathbb{E} \in \mathbb{E}^V : I_\Phi(\mathbb{E}) \geq \gamma\}$ ).

Es precisamente en este momento cuando podemos poner el límite entre los dos extremos y buscar una solución de compromiso tal que: el factor  $\gamma$  debe ser lo suficientemente alto como para asegurar un valor de inmersión válido y una validez ecológica aceptable, pero manteniendo el compromiso de la computabilidad *asequible*. Este término *asequible* nos da la clave para determinar que ese concepto de asequibilidad variará según evolucione la tecnología.

Comencemos entonces por estudiar la física tal y como se enuncia hoy en día.

### 3.4.4 Principios físicos modernos

En la física moderna se identifican tres fuerzas fundamentales:

- *Fuerzas nucleares*
- *Fuerzas electromagnéticas*
- *Fuerzas gravitatorias*

Aunque la existencia de estas interacciones fundamentales es incontestable y su implementación en motores físicos computacionales aumentaría enormemente la validez ecológica de los entornos virtuales, no existen software en la actualidad que lo permita. Es necesario encontrar un nivel de abstracción suficientemente alto como para conseguir una similitud con los servicios que vamos a encontrar en el software simulador de física actual. Aunque puede que dentro de algunos años si que existan simuladores capaces de realizar una simulación a nivel molecular o incluso de partículas subatómicas, a día de hoy estamos muy lejos de poseer esa capacidad de cómputo.

Por eso aunque los servicios arriba indicados son válidos desde un punto de vista físico, no nos serán todos directamente útiles, excepto la fuerza gravitatoria que denominaremos  $\Psi^{\mathfrak{R}} \mapsto G$ .

Buscando la optimización desde el punto de vista de coste computacional vamos a definir los servicios base de entorno real  $\Psi^{\mathfrak{R}}$  como:

- $\Psi^{\mathfrak{R}} \mapsto NP = \text{Posición}$

- $\Psi^{\mathfrak{N}} \mapsto NV = \text{Velocidad}$
- $\Psi^{\mathfrak{N}} \mapsto NA = \text{Aceleración}$
- $\Psi^{\mathfrak{N}} \mapsto NM = \text{Masa}$

### 3.4.5 La persistencia del entorno

Aunque la persistencia en el mundo real es algo tan natural que no somos conscientes de ella, en un mundo virtual no es un problema sencillo. La persistencia también está muy arraigado al principio de causalidad. Si dentro de un entorno virtual dejamos un objeto estático en un lugar, ese objeto debe mantenerse en el mismo sitio. Aunque el software de simulación se reinicie, ese objeto deberá permanecer estático hasta que otra interacción lo altere, al igual que ocurre en el mundo real.

Es necesario por lo tanto definir la persistencia como un servicio

- $\Psi^{\mathfrak{N}} \mapsto P = \text{Persistencia}$

### 3.4.6 La impenetrabilidad como principio de la interacción

La interacción entre elementos surge como consecuencia de las fuerzas electromagnéticas. El principio básico de la física macroscópica de la «impenetrabilidad», aunque corroborada por el *Principio de Exclusión de Pauli* a nivel cuántico, es perfectamente entendible intuitivamente. Se resume como: «ningún cuerpo puede ocupar al mismo tiempo el lugar de otro/la resistencia que opone un cuerpo a ser traspasado». A la escala en la que los seres humanos percibimos la realidad y a la que la *metáfora del entorno* tiene validez, esta propiedad física es la base de la interacción.

- $\Psi^{\mathfrak{N}} \mapsto I = \text{Impenetrabilidad}$

Haciendo resumen de los servicios que vamos a extraer de la realidad física que nos rodea, estos son los finalmente elegidos:

- $\Psi^{\mathfrak{N}} \mapsto G = \text{Gravedad}$
- $\Psi^{\mathfrak{N}} \mapsto N.P = \text{Posición newtoniana}$

- $\Psi^{\mathfrak{N}} \mapsto N.V = \text{Velocidad newtoniana}$
- $\Psi^{\mathfrak{N}} \mapsto N.A = \text{Aceleración newtoniana}$
- $\Psi^{\mathfrak{N}} \mapsto N.M = \text{Masa newtoniana}$
- $\Psi^{\mathfrak{N}} \mapsto P = \text{Persistencia}$
- $\Psi^{\mathfrak{N}} \mapsto I = \text{Impenetrabilidad}$

### 3.4.7 Identificación de servicios Perceptivos

Además de basarnos en la realidad para inducir qué servicios físicos existen, debemos hacer lo propio con el depositario último de toda simulación: la percepción humana ( $PH$ ).

Los servicios perceptivos son aquellos encargados de transferir información sensorial al sujeto. Estos servicios son obligatorios ya que sin ellos los **Entornos Virtuales Inmersivos** no son directamente experimentables y por lo tanto no serían inmersivos.

Dentro de los servicios perceptivos haremos una distinción importante dependiendo si la información perceptiva es externa o interna al individuo. En el caso de ser información externa se denominará sensorial, y en el caso de información interna se denominará propiocepción.

- $PH \mapsto S = \text{Sensoriales}$
- $PH \mapsto P = \text{Propioceptivos}$

Serán los servicios sensoriales aquellos asociados a los estímulos visuales, auditivos y hápticos, mientras que los propioceptivos se dividirán en los servicios de autoimagen e inventario.

Dentro de los servicios perceptivos  $\Phi$  existe una taxonomía anidada de nivel 2 que denominaremos Sensorial y consta de tres subniveles:

- $PH \mapsto S.V = \text{Percepción visual}$
- $PH \mapsto S.A = \text{Percepción auditiva}$
- $PH \mapsto S.H = \text{Percepción háptica (táctil, olfativa o del gusto)}$

Por otro lado los servicios propioceptivos, al referirse a la autoconciencia que todos los humanos disponemos de nosotros mismos, podríamos sumergirnos en múltiples consideraciones psicológicas y filosóficas. Pero como el objetivo de este modelo es simplificar y dar una visión orientada a la simulación vamos a reducirlo a dos servicios: autoimagen e inventario. En otras palabras, para tener una experiencia realista en un entorno virtual debemos poder contar con una imagen personal y reconocible para identificarnos a nosotros mismos «dentro» del entorno. Por otro lado debemos contar con la posibilidad de poseer y almacenar elementos dentro del entorno para poder desarrollar apego al entorno.

- $PH \mapsto P.A = \text{Autoimagen}$
- $PH \mapsto P.P = \text{Inventario}$

### 3.4.8 Identificación de servicios Interactivos

Anteriormente hemos investigado el entorno físico, posteriormente hemos buscado dentro de la naturaleza humana buscando características necesarias. Por último vamos a explorar cómo los seres humanos interactuamos con nuestro medio para extraer los requisitos mínimos que un entorno virtual ha de cumplir.

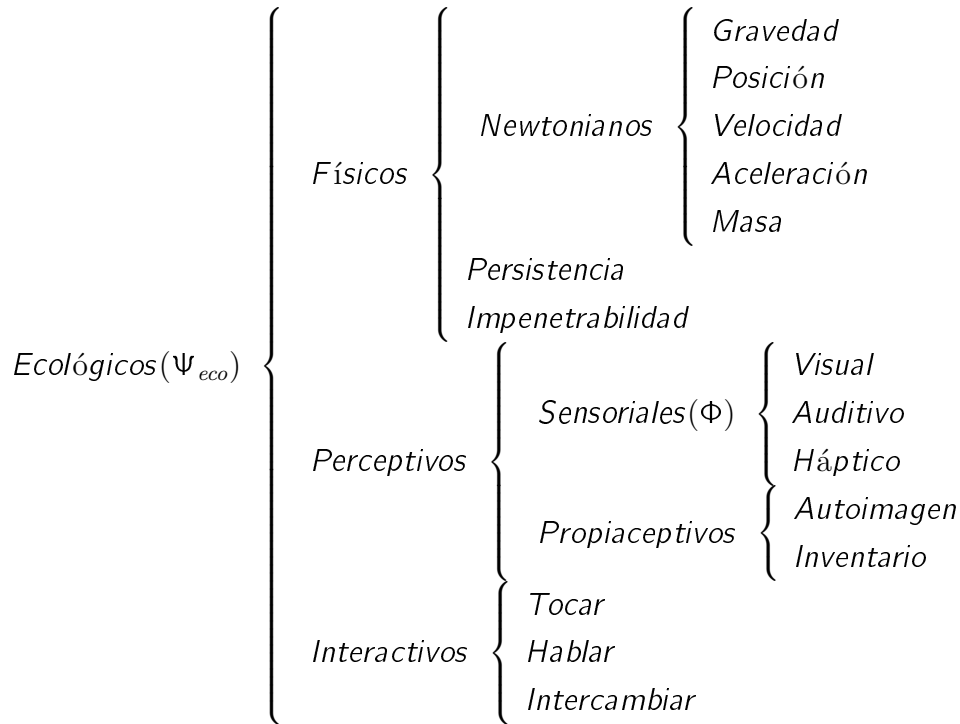
Si nos fijamos en las múltiples maneras en las que podemos interactuar con nuestro entorno veremos que en mayor o menor medida todas pueden ser resumidas con el hecho de tocar, hablar o intercambiar elementos. En «hablar» agruparemos todas las posibles maneras de intercambiar información, en contraposición de «intercambiar elementos» que simplemente definiremos intercambiar.

- $INTERACTUAR \mapsto I.T = \text{Tocar}$
- $INTERACTUAR \mapsto I.H = \text{Hablar}$
- $INTERACTUAR \mapsto I.I = \text{Intercambiar}$



### 3.4.9 Taxonomía completa de Servicios Ecológicos

Resumamos en este punto la taxonomía completa tal y como la hemos definido hasta el momento.



### 3.4.10 Formalización notación taxonómica

Ya hemos avanzado enormemente en la categorización de los servicios ecológicos, pero dada la amplitud del árbol de dependencias se hace necesario definir una notación que nos permita trabajar de una manera cómoda con el modelo.

Recordando el modelo matemático propuesto en el punto 3.3, los servicios  $\Psi$  representan toda la capacidad de cómputo e interactiva de un entorno. Como se ha definido en el punto 3.3.1.3 los *servicios* forman parte de un *entorno* y no por lo tanto no es posible referenciar  $\Psi$  sin estar asociado a un entorno  $\mathbb{E}$ . Excepto en el caso en el que nos estemos refiriendo al conjunto universal de todos los servicios existentes en todos los entornos posibles. En este caso abstracto, hablaremos de  $\Psi$  «estrella» y se definirá como:

$$\Psi^* = \{\psi_1, \psi_2, \dots, \psi_n\} / \psi_i \in \mathbb{E}^*$$

Incluir todas las posibles maneras de interactuar en una sola categoría es muy útil a la hora de formalizar el modelo, pero para poder utilizarlo como base de trabajo es necesario presentar una taxonomía y algunos casos especiales.

Para ayudarnos en esta tarea vamos a definir el operador  $\odot$  como la expresión de pertenencia relativa un servicio a una taxonomía concreta. Por ejemplo la expresión «el servicio  $\Psi$  es de tipo Perceptivo» se expresa como

$$\Psi \odot Ecológico.Perceptivo$$

y lo expresaremos siguiendo esta nomenclatura

$$\Psi_{tax_1, tax_2 \dots tax_n} / tax_i \equiv taxonomía\ nivel\ i$$

Pero si utilizásemos todo el nombre de la categoría podría resultar en una notación farragosa. Para evitarlo utilizaremos abreviaturas basándonos en los n-primeros caracteres diferenciadores del nombre del servicio. De esta manera tendríamos:

$$\Psi \odot E.P \implies \Psi_{E.P}$$

Por ejemplo podríamos referirnos a todos los servicios Hápticos de un entorno mediante esta nomenclatura como  $\Psi_{E.P.S.H}$

Existen algunas excepciones a esta nomenclatura en la que vamos a utilizar un único símbolo para identificar ciertos subconjuntos taxonómico muy utilizado:

- $\Psi_{\Phi} \equiv \Psi_{E.P.S} \equiv \text{«Servicios Inmersivos»}$
- $\Psi_{eco} \equiv \Psi_E \equiv \text{«Servicios Ecológicos»}$

### 3.4.11 Tecnología

Hasta ahora hemos estudiado los servicios denominados «ecológicos» como aquellos que son naturales al ser humano y hemos conseguido disponer de un conjunto de servicios que nos aseguran la capacidad de experimentar un entorno virtual siempre que cumpla con algunas condiciones. Disponemos de una base sólida sobre la que experimentar.

Tal y como hemos argumentado la definición de servicios ecológicos y tecnológicos y, partiendo de que hemos definido completamente los *eco* podremos definir los servicios tecnológicos como «aquellos servicios que no son ecológicos»

Para abreviar denominaremos *tec* a los «servicios tecnológicos» de tal manera que se definan

$$\Psi_{tec} = \overline{\Psi_{eco}}$$

Para definir  $\Psi_{tec}$  de manera más gráfica, dependerán de la implementación final del entorno virtual y del conjunto de objetivos que persiga.

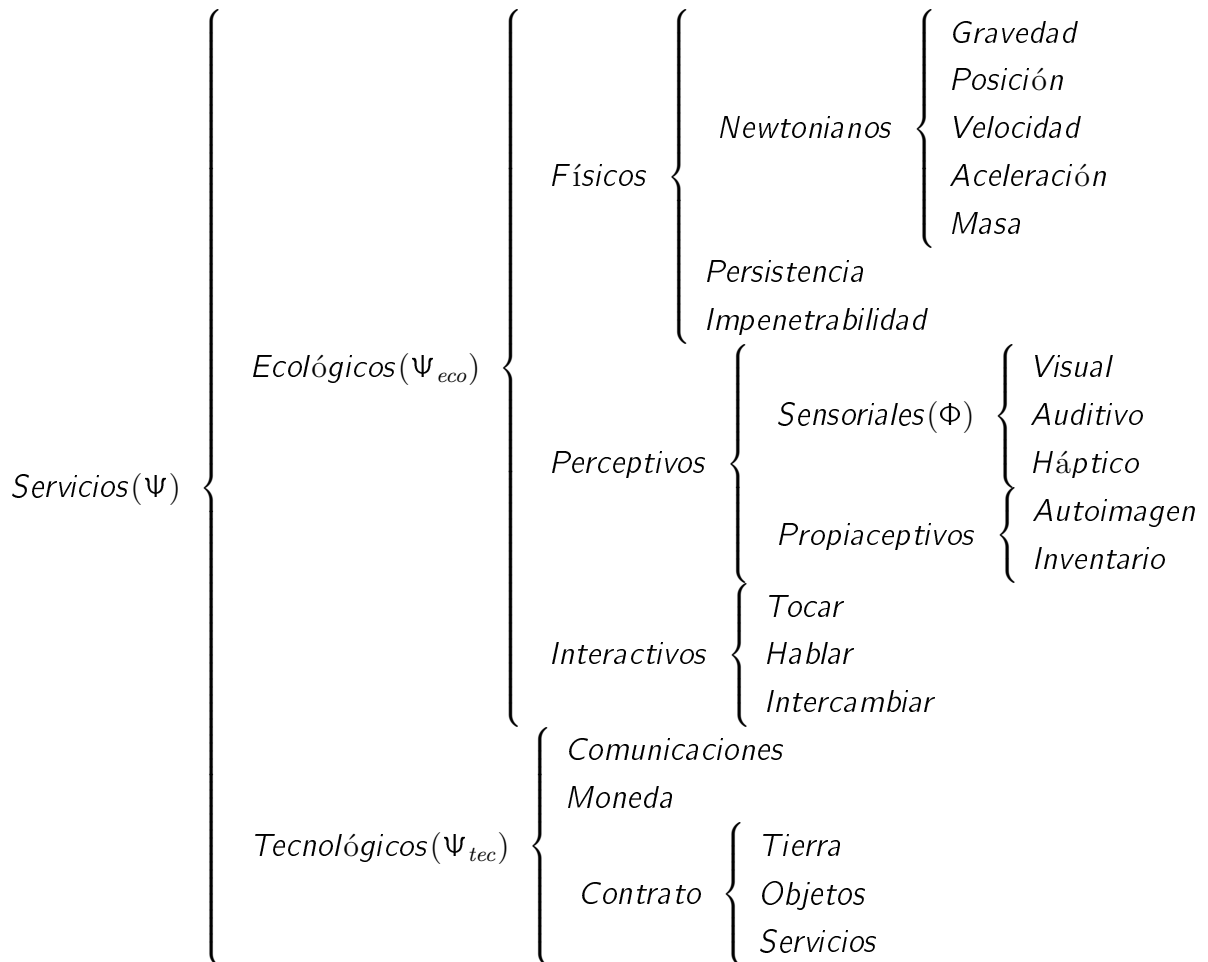
En el mundo real existen multitud de servicios que son ofrecidos por la tecnología. De hecho la tecnología se define como «[...] *el conjunto de conocimientos técnicos, científicamente ordenados, que permiten diseñar y crear bienes, servicios que facilitan la adaptación al medio ambiente [...]*», es decir son una ayuda para que los seres vivos nos adaptemos mejor al entorno.

Como es obvio en el mundo real toda la tecnología viene explicada por reacciones físicas de mayor o menor nivel (reacciones químicas, electromagnetismo, propiedades físicas de los elementos, etc.). Sin embargo habitualmente los humanos encapsulamos esas reacciones en servicios tecnológicos de alto nivel. Por ejemplo la gran mayoría de las personas que sabe conducir un coche ni sabe ni le interesa saber cómo funciona un motor de combustión interna o se transmite la fuerza desde el cigüeñal a los neumáticos. Se ha encapsulado ese conocimiento en un servicio de nivel alto que responde a un interfaz de usuario más o menos complejo. Esta realidad ocurre constantemente y se replica en multitud de elementos cotidianos.

Por lo tanto, y al contrario de lo definido con los servicios ecológicos, vamos a plantear los servicios tecnológicos como un conjunto abierto y en constante expansión. Como base se proponen un conjunto reducido que suelen estar implementados en los *proveedores de entorno* más comunes que se detallarán en el punto 3.10 : *comunicaciones, moneda, contratos de tierras, objetos y servicios*.

### 3.4.12 Taxonomía de Servicios, conclusiones

Como resumen a esta sección se representa aquí el esquema completo de los servicios identificados.



## 3.5 Modelo arquitectónico DEC2RPM

Una vez hemos definido el modelo teórico y una taxonomía de servicios es el momento de volver a aplicar el paradigma de la metáfora del entorno a los esquemas que hemos definido con el objetivo de plantear un modelo arquitectónico que sea capaz de soportar e implementar soluciones basadas en *Entornos Virtuales Inmersivos*. Para ello exponemos en este punto el modelo teórico arquitectónico denominado «**DEC2RPM**» y que se corresponde con las siglas de las palabras «*Renderer, Controller, Deployment, Representation, Physical y Engine*» tras ser reordenadas y optimizadas mnemotécnicamente.

En primer lugar vamos a definir un simple experimento imaginario de comportamiento que nos permitirá identificar tres importantes pilares de la arquitectura.

### 3.5.1 El experimento del comportamiento complejo

Los seres humanos percibimos dos tipos de comportamiento complejo: el comportamiento inteligente o una conjunción simultánea de acciones sencillos. Pero ¿qué podemos definir como *comportamiento complejo*? Definamos un comportamiento complejo como aquel que no se puede explicar en función de la aplicación de reglas simples y conocidas. Para caracterizar esta afirmación vamos a realizar una serie de experimentos.

Primero, situándonos en el mundo real y bajo una escala física antropomórfica en la que la metáfora del entorno tenga validez, tomaremos un objeto, por ejemplo una esfera. Si soltamos la esfera desde cierta altura, entendemos que la gravedad hará que se desplace en el sentido inverso de su eje Z hasta tocar el suelo. Esto no sería un comportamiento complejo ya que sería lo esperado y para su implementación basta con aplicar la gravedad al objeto.

En el segundo experimento, siguiendo el mismo principio, soltaremos la esfera desde cierta altura. Esta vez en se moverá en el sentido positivo de su eje Z. Nuestro cerebro humano nos hará pensar que dicha esfera o bien es capaz de flotar o bien hay algo que está tirando de ella hacia arriba. Si bien este segundo experimento el comportamiento es algo más complejo que en el primero, la complejidad será consecuencia de la interacción con un número mayor de artefactos que interactúan. Podemos extender este segundo experimento añadiendo más y más interacciones y artefactos hasta llegar a un sistema no lineal. Tal y como se argumenta mediante la teoría del caos «... pequeñas variaciones

en dichas condiciones iniciales pueden implicar grandes diferencias en el comportamiento futuro, imposibilitando la predicción a largo plazo. . . ». Este tipo de comportamiento sí que puede llegar a ser complejo pero habitualmente entrará dentro de los servicios  $\Psi_{E.F}$  en nuestro modelo, al ser habitualmente implementados por motores físicos.

En un tercer experimento, dado el mismo estado inicial, al soltar la esfera, veremos que ésta cae al suelo y sigue rodando esquivando algunos objetos para finalmente detenerse detrás de un obstáculo suficientemente grande como para que nos bloquee su visión directa. Al movernos para poder seguir observando la esfera del experimento, ésta volverá a moverse hasta salir de nuestro campo de visión. ¿Qué conclusiones extraería nuestro cerebro de este comportamiento? A diferencia de los experimentos primero y segundo, nuestra comprensión antropomórfica del entorno nos dice que esa esfera ha sido afectada por las leyes físicas al caer al suelo y rodar (hasta aquí nada fuera de lo común), pero que también ha sido capaz de percibir su entorno y actuar en consecuencia. Más aún, ha demostrado una intención al ocultarse de nuestra vista lo que también intuimos que subyace a un estado mental que posiblemente podríamos clasificar como “miedo”.

En resumen, en el primer experimento no encontramos un comportamiento complejo, mientras que en el segundo y tercero sí. Sin embargo los tipos de complejidad son distintos, mientras que en el segundo la complejidad se desprende del gran número de interacciones y elementos que realizan acciones de complejidad baja, el tercero refleja un potencial de comportamiento inteligente.

Aplicando los experimentos realizados a la búsqueda inductiva de servicios que estamos realizando en este punto podemos extraer la conclusión que nuestro modelo deberá contemplar interacciones complejas de origen tanto físico como inteligente.

### 3.5.2 Interacciones físicas complejas

En cualquier entorno virtual existirán un número indeterminado de interacciones físicas entre objetos. Parece lógico suponer que a mayor tamaño de entorno y mayor realismo estas interacciones físicas serán más abundantes y complejas pudiendo encontrarnos con colisiones múltiples en cadena y eventualmente simulación de sistemas no lineales. Este tipo situaciones se resuelve habitualmente mediante un tipo concreto de software denominado generalmente “Motor de Física”.

La complejidad de este tipo de software es muy alta, conformando un campo de

estudio en sí mismo y escapándose del alcance de la presente tesis. Independientemente de este hecho, es obvio que los servicios que hemos categorizado como  $\Psi_{E.F}$  deberán ser implementados a través de un “motor de física”, y por lo tanto cualquier entorno virtual que intente tener un mínimo valor ecológico deberá utilizar un software de este tipo. Por esta razón vamos a definirlo como un pilar básico en el modelo teórico propuesto.

Todas las interacciones físicas entre elementos se implementarán en los servicios  $\Psi_{E.F}$ . La complejidad final de la interacción dependerá del escenario virtual que se diseñe: número de elementos, potencial de interacción entre ellos, etc. Obviamente esta complejidad estará limitada por la bondad del motor de física y los recursos hardware dedicados.

### 3.5.3 Interacciones inteligentes

Como se ha demostrado en el tercer experimento, un motor de física no es suficiente para implementar un comportamiento que denominaremos “inteligente”. En este punto podríamos reflexionar sobre la necesidad o no de este comportamiento en un entorno virtual, ya que es posible que para algunos tipos de entorno no sea necesario.

Este sería un caso particular de entornos tipo no-interactivo o de interactividad únicamente física. Según la taxonomía descrita este podría tener un índice inmersivo alto, pero un índice ecológico muy bajo. Pero dado que nuestro objetivo es definir un modelo teórico formal y generalista es imperativo contemplar la existencia de esta necesidad.

Este *comportamiento inteligente* que estamos buscando únicamente puede provenir de dos fuentes: inteligencia natural o inteligencia artificial. Para abreviar estos dos términos en el futuro nos referiremos a ellos como **IN** e **IA** respectivamente.

En el caso de la **IN** el entorno debe implementar servicios que permitan a los seres humanos interactuar entre ellos a través de sus avatares y con los servicios de comunicación. Habitualmente los seres humanos aparecerán dentro del simulador como *avatares* que no es más que el nombre de un objeto particular que representa la personificación de un agente inteligente externo.

En el caso de la inteligencia artificial llegamos de nuevo a un campo de estudio muy amplio, sin embargo en este caso sí que vamos a definir y proponer una solución.

Por lo tanto debemos definir dos módulos adicionales encargados de incluir la posibilidad de comportamiento e interacciones inteligentes: un módulo de acceso para

inteligencias «naturales» **IN** y otro para inteligencias «artificiales» **IA**.

### 3.5.4 El concepto de Avatar

Aunque ya se ha comentado el concepto con anterioridad en el capítulo 2 es interesante reflexionar brevemente sobre su significado real. En términos sencillos, un avatar es la forma clásica en la que se representa a un *usuario* dentro de un entorno virtual. Con el término *usuario* nos estamos refiriendo a una persona (**IN**) que, mediante el uso de algún tipo de tecnología de soporte, es capaz de disfrutar de una representación visual, auditiva y/o háptica de un entorno virtual.

Para darle una definición más completa acorde a nuestro modelo plantearemos «avatar» como objeto o grupo de objetos que, dentro de un entorno virtual, representan a un usuario de manera clara y aparente. Es decir, un avatar lo es tanto en cuanto el resto de usuarios pueden identificarlo claramente como avatar.

Sin embargo, y aunque pueda parecer lo contrario, no existe una relación biunívoca entre «avatar» y «usuario». Es decir, un usuario puede percibir un entorno virtual sin la necesidad de un avatar. Al mismo tiempo un usuario podría disponer de una serie de avatares distintos preparados para su uso, incluso de manera simultánea, en un mismo entorno virtual. La cardinalidad de la relación «avatar» y «usuario» se define como:

$$\boxed{\text{USUARIO}} \begin{smallmatrix} 1 \\ 1 \end{smallmatrix} \longleftrightarrow \begin{smallmatrix} N \\ 0 \end{smallmatrix} \boxed{\text{AVATAR}}$$

Por otro lado es técnicamente posible que un avatar sea manejado por una **IA**, aunque trataremos las distintas posibles maneras en las que la inteligencia artificial pueda participar en un entorno virtual más adelante en este capítulo. En cualquier caso, ya sea una **IN** o una **IA** quien esté manejando un avatar se aplicarán los mismos términos en el presente modelo.

Tal y como se ha definido la cardinalidad, un usuario no requiere un avatar para percibir un entorno virtual. Sin embargo ese mismo usuario necesitará los avatares del resto de usuarios para poder percibirlos como tal. Se establece entonces una relación reflexiva que resume la idea «la percepción que el resto tiene de mí». Es por tanto obvio asociar la necesidad de un avatar a un entorno social, interactivo y antropomórfico.



Si enfocamos este concepto de «avatar» desde el punto de vista de la taxonomía de servicios del punto 3.4 se implementará a partir de  $\Psi_{E.P.P.}$ , es decir los servicios «propioceptivos». No es por tanto posible definir un modelo arquitectónico que sea capaz de tener un  $I_{eco}$  (índice de validez ecológica) alto sin que exista un módulo encargado de implementar esta gestión de «avatares».

### 3.5.5 Taxonomía de interacciones inteligentes entre elementos según su naturaleza

Vamos a estudiar a continuación qué tipos de interacciones inteligentes vamos a poder encontrarnos dependiendo de la naturaleza de la inteligencia que está detrás de los miembros de cada interacción.

Ya se ha sentado la base del concepto de avatar según la cual siempre que se utiliza este vocablo nos estamos refiriendo a una representación virtual de un ser humano o de una **IA** que pretende ser un ser humano. Si ese fuese el caso no existirá una certeza absoluta que nos permita diferenciarlos así que vamos a asumir que todo avatar representa a un usuario, o lo que es lo mismo a una inteligencia natural **IN**. Ahora vamos a dividir todos los objetos de un entorno en dos clases, avatar o no-avatar, para poder estudiar los distintos tipos de interacción posible.

$$\forall \omega \in \Omega : \nexists \omega \in (\Omega_{avatar} \cap \Omega_{no-avatar})$$

$$\Omega_{avatar} = \overline{\Omega_{no-avatar}}$$

Existen por tanto tres combinaciones posibles, que se pasan a detallar a continuación:

- **Avatar**  $\iff$  **Avatar** : Este es el caso clásico en el que dos usuarios hablan, se miran, se tocan o intercambian objetos. Aunque no tengamos certeza absoluta que todo avatar sea controlado por un usuario, en ese caso la **IA** deberá comportarse como un usuario y seguir el mismo canal de interacción, por lo que no habrá diferencia.
- **Avatar**  $\iff$  **No-avatar** : Un usuario toca, empuja, golpea, habla a, coge o cualquier otra acción posible a un objeto no-avatar. Así mismo el objeto no-avatar podrá hablar, tocar o realizar el mismo tipo de interacciones posibles. Para hacer

un paralelismo con el entorno real siguiendo la metáfora del entorno: una persona utilizando un teléfono o una persona jugando con un cachorrito.

- **No-avator**  $\iff$  **No-avator** : Por último, dos objetos no-avatares pueden interactuar entre ellos de manera análoga. Si bien es cierto que en estos casos no será necesario utilizar el lenguaje humano, no existe un límite de lo que se puede usar o no. Un claro ejemplo en el entorno real es el de un smartphone enviando una imagen a otro smartphone.

De estas tres combinaciones en realidad extraemos dos tipos diferenciados: si hay un avatar involucrado o no. En el momento que un usuario está involucrado en la interacción se ha de utilizar un canal acorde a su capacidad de comunicación.

### 3.5.6 Los tres canales de interacción

Recapitulando sobre las taxonomías y experimentos realizados en las últimas secciones vamos a definir en nuestro modelo tres posibles niveles de interacción: nivel físico, nivel no-avator y nivel avator. Para simplificar la nomenclatura y para expresar que el nivel de complejidad de dichas interacciones es acumulativa vamos a designarlos respectivamente como  $ch\alpha$ ,  $ch\beta$  y  $ch\gamma$ :

- $ch\alpha$  : Este es el nivel más bajo implicando únicamente los objetos del entorno y el *motor de física*. En el nivel  $\alpha$  los objetos interactúan entre ellos sin entrar en detalles sobre si son avatares o no, aplicando lo más fielmente posible las reglas físicas del entorno real, implementadas por el *motor de física*. La información que circule por este canal serán las iteraciones secuenciales de la entrada y la salida del *motor de física*, por lo que dicha información no estará estructurada.
- $ch\beta$  : Este canal será el utilizado para las interacciones **IA**  $\iff$  **IA**. Aunque el nivel de abstracción será más alto que en  $ch\alpha$  al estar estructurada y orientada al servicio, tanto la velocidad como el contenido no será apto para la comprensión humana directa. El módulo equivalente al *motor de física* de  $ch\alpha$  para  $ch\beta$  será denominado *despliegue-representación*.
- $ch\gamma$  : Este es el canal con más alto nivel de abstracción. La información que transita está estructurada y orientada a los servicios inmersivos  $\Psi_\phi$ . En otras palabras,

este canal estará compuesto por una secuencia constante de imágenes, sonidos e información háptica capaces de estimular los sentidos del usuario. El módulo equivalente al *motor de física* de  $ch_\alpha$  y para *despliegue-representación*  $ch_\beta$  será denominado *interprete-controlador*.

En esta representación visual del modelo de canales se puede observar cómo la cantidad de información del entorno transmitida es menor según se asciende por la escala, así como su especificidad y destino.

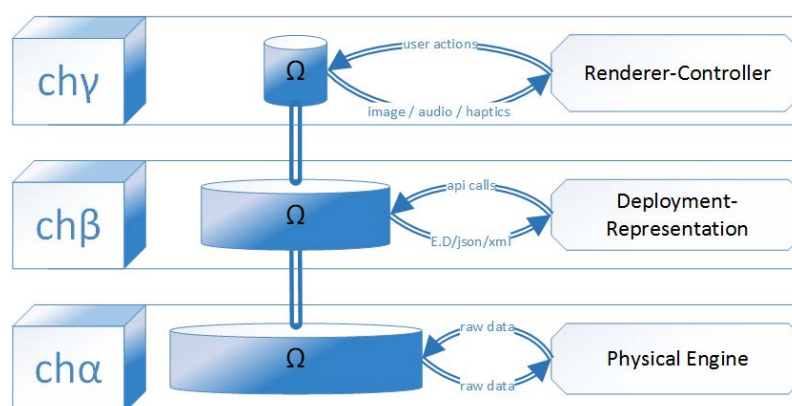


Figura 3.1: Diagrama de Canales

Podemos por tanto definir en este momento que el modelo arquitectónico que proponemos está dividido en tres módulos de primer orden: *motor físico*, *despliegue-representación* e *interprete-controlador*. A continuación vamos a describirlos en profundidad.

### 3.5.7 Módulo Motor Físico

Como ya se ha comentado en el punto 3.5.2 un *motor físico* o *motor de física* es un tipo de software específico (Erleben, 2004) que, dado un conjunto de datos que representan objetos en una geometría dada es capaz de reproducir los efectos de unas leyes físicas preestablecidas. De manera prácticamente universal hasta la fecha, todos los *motores físicos* existentes se centran en un entorno euclídeo de dimensión 3 (espaciales) o dimensión 2 (plano). La complejidad de este tipo de software es muy alta, así como su consumo de CPU sobre todo en 3D.

La mayor parte de estos motores han sido desarrollados en la implementación de software recreativo. Este hecho no es casual ya que en muchas ocasiones un juego es, aunque incompleto, un entorno virtual.

A pesar de que existe una gran variedad de *motores físicos* (Cowan and Kapralos, 2017; Nevelsteen, 2015; Renz and Ge, 2017) todos comparten una arquitectura similar: entrada de datos espaciales, procesamiento y salida de datos espaciales que vuelven a ser re-alimentados en un bucle infinito.

Como ya se ha comentado anteriormente la complejidad de este tipo de software es muy grande y queda fuera del alcance de esta tesis. Es importante recalcar sin embargo que el motor físico es una pieza *sine qua non* en un entorno virtual con una mínima seriedad. Un entorno virtual sin simulador físico carecería de leyes físicas y por lo tanto los índices de inmersión y ecológico se verían seriamente perjudicados.

### 3.5.8 Módulo Interprete - Controlador

El canal *chγ* identifica uno de los módulos de primer orden del modelo, el denominado «intérprete - controlador». Este módulo software se identifica con el «interfaz de usuario» ya que es capaz de interpretar la información geométrica vectorial del entorno y generar imágenes sintéticas del mismo.

Al igual que ocurre con los *motores físicos* en el campo del **CGI** *computer generated images* existen numerosos *renderers* o *motores gráficos* ampliamente utilizados en el campo del arte y el ocio. Y de igual manera, la complejidad de este tipo de software es muy grande, quedado también fuera del alcance de esta tesis.

Por último, este módulo no únicamente debe ocuparse de la generación de información visual, sino también auditiva y háptica. Además debe ser capaz de recoger información de vuelta desde el usuario y poder inyectarla en el entorno. Esta información estará formalizada en acciones como: moverse, tocar, hablar o influir en el resto de objetos del entorno.

Este módulo también es fundamental para un entorno. Simplemente, si los seres humanos no podemos percibir directamente un entorno el índice de inmersión será cero.

### 3.5.9 Módulo Despliegue - Representación

Podríamos encontrar muchos casos de entornos virtuales en los que el módulo de despliegue - representación no existe o es implementado parcialmente de manera enmascarada dentro de alguno de los otros dos. Por ejemplo: un entorno virtual en el que se recrea la realidad

sin ningún tipo de tecnología o fauna. Los objetos podrían caer, ser empujados y respetar las leyes Newtonianas y los avatares podrían hablar entre ellos pero no sería posible hacer nada más.

La evolución histórica ha obviado este módulo habiendo quedado eclipsado por la importancia de los otros dos y habitualmente se ha suplido su existencia aplicando técnicas de *scripting* a nivel de *motor físico*. Sin embargo cuando se intenta implementar un sistema siguiendo la metáfora del entorno es palpable que no incluir este módulo es un error de diseño que coarta el potencial del mismo.

A diferencia del módulo *motor físico* y del módulo *intérprete - controlador*, definir en profundidad módulo *despliegue - representación* sí que es el objetivo de esta tesis y será ampliado en la sección 3.6 y ampliamente expuesto a lo largo del capítulo 3.7, pero vamos a adelantar algunos de sus detalles en este momento.

Tal y como ya se ha comentado en la sección 3.2 proponemos los *Sistemas Multiagente Simulados* en el punto 3.12 como la mejor propuesta tecnológica para implementar técnicas de comportamiento inteligente artificial dentro de *entornos virtuales inmersivos*. Sin embargo no existe aún un modelo teórico sobre el cual diseñar y posteriormente construir estos entornos virtuales con garantías de éxito, teniendo en cuenta indicadores matemáticos como el índice de inmersión.

### 3.5.10 Módulo de Soporte de Sistema

Por último existe un cuarto módulo con mucha menos entidad funcional que el resto pero necesario para la coordinación del resto. Dependiendo de las características finales de la tecnología elegida para la implementación del sistema, este módulo podrá tener más o menos envergadura.

### 3.5.11 Modelo arquitectónico, conclusiones

En resumen existirán tres módulos de orden principal y uno de apoyo. Sus nombres y siglas en inglés serán, respectivamente: Physical Engine (PEM), Deployment Representation Model (DRM), Renderer-Controller (RCM) y System Support (SSM).

Cada módulo poseerá, o podrá poseer, su propio almacén de datos, pero además todos los datos del entorno serán accesibles por todos los módulos, en función del canal

que se utilice.

Por último, existirán dos actores: los usuarios finales y el sistema de comportamiento inteligente que estarán servidos a través del RCM y el DRM respectivamente.

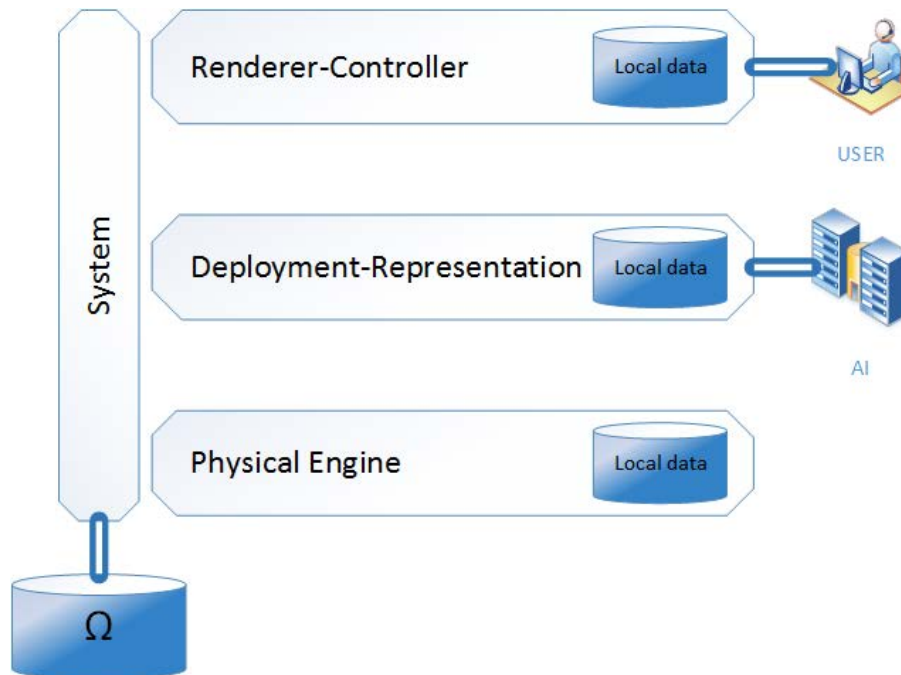


Figura 3.2: Diagrama de Módulos

## 3.6 Modelo Despliegue - Representación

En la presente sección vamos a definir los parámetros teóricos del *módulo despliegue - representación* identificado en el punto 3.5.9. Este modelo será denominado **Modelo Despliegue-Representación** o **DRM** de manera abreviada y es una parte fundamental del modelo **DEC2RPM**.

Dado que el objetivo en esta sección es ceñirnos al modelo teórico no se van a exponer situaciones concretas ni estrategias de implementación en este punto, esto se tratará en el capítulo 3.7 donde se definirá la metodología propuesta asociada al modelo.

### 3.6.1 Objetivos marcados al DRM

A la hora de definir el **DRM** desde un punto de vista teórico nos planteamos los siguientes requisitos a ser cumplidos:

- **Metáfora del Entorno:** El modelo seguirá de la manera más fiel posible los preceptos propuestos para seguir la *Metáfora del Entorno* definida en el punto 3.2.
- **Índice Inmersivo:** El modelo estará optimizado para funcionar en entornos con un Índice Inmersivo ( $I_{\Phi}$ ) maximizado, según se define en el punto 3.3.4
- **Índice de Validez Ecológica:** Así mismo el modelo estará también optimizado para funcionar en entornos con un Índice de validez ecológica ( $I_{eco}$ ) maximizado.
- **Multientorno:** Introducimos aquí el concepto de *simulación paralela y multientorno* refiriéndonos a la posibilidad de que un mismo sistema **DRM** de soporte a distintos entornos virtuales o incluso a entornos virtuales y al entorno real.

### 3.6.2 Contexto

El **Modelo Despliegue-Representación**, haciendo referencia a (Wooldridge and Jennings, 1995), permitirá maximizar la **coherencia** y la **coordinación** de un **Sistema Multi Agente Situado** Para comprender adecuadamente el contexto en el que es válido este modelo, ha de ser tenido en cuenta que ha sido diseñado específicamente para modelar **Sistemas Multi Agente Situados** en **Entornos Virtuales Inmersivos**. Aunque en capítulos

anteriores ya hemos hablado de estos dos conceptos de manera exhaustiva vamos a refrescarlos mediante estas breves definiciones:

**Sistema Multiagente Situado:** Sistema Multiagente cuyos agentes se desenvuelven en un entorno espacial. Cuando dicho entorno espacial posee las tres dimensiones físicas espaciales (X-Y-Z) más la dimensión temporal (T) hablamos de Sistema Multiagente 3D. Para abreviar utilizaremos la nomenclatura **M.A.S**

**Entorno Virtual Inmersivo:** Entorno virtual que es habitualmente implementado a través de un software de simulación y posee la capacidad de ser experimentado a través de distintos medios físicos de una manera inmersiva. Estos entornos están íntimamente ligados a la tecnología normalmente denominada *Realidad Virtual*. Para abreviar utilizaremos la nomenclatura **V.I.E**

Es importante reseñar que **DRM** no es una implementación de software concreta, ni una solución basada en componentes, es un modelo que define una arquitectura genérica para un entorno específico. Por ejemplo, siguiendo el modelo **DRM** podríamos construir distintas aplicaciones perfectamente válidas pero que no compartan una sola línea de código.

### 3.6.3 División entre Despliegue y Representación

El **DRM** se divide en dos submódulos que denominaremos «Estructura de Despliegue» y «Representación Simbólica Espacial» respectivamente. Esta separación de módulos responde a la necesidad de disponer de una capa de abstracción del entorno para poder encapsular los servicios internos y externos a cada entorno.

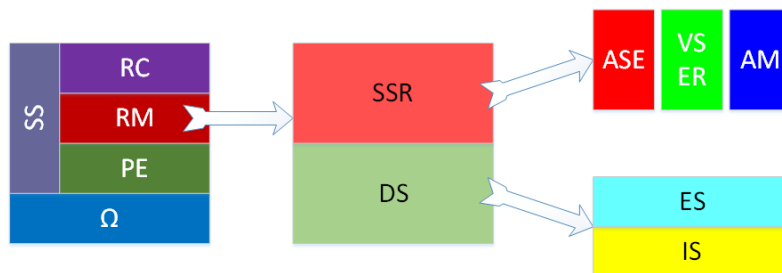


Figura 3.3: Diagrama Teórico del modelo DRM de alto nivel



Otra razón para hacer esta división está especialmente orientada a cumplir con el objetivo *Multientorno*, para lo cual es necesario plantear una capa de abstracción a nivel de entorno de manera que los servicios se dividan en internos y en externos. Esta característica también la denominaremos *Simulación Paralela* para reflejar la potencia de disponer de una sola arquitectura que sea utilizable en distintos entornos.

Este es un diagrama de ejemplo de esta arquitectura paralela:

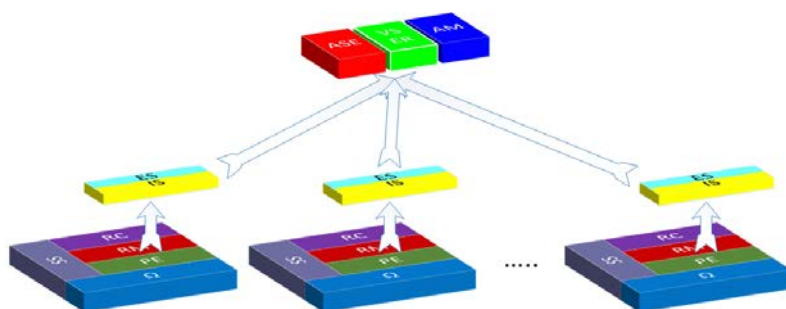


Figura 3.4: Arquitectura en Simulación Paralela

### 3.6.4 Estructura de Despliegue

Desde un punto de vista teórico, la estructura de despliegue será donde se implementen los servicios tecnológicos de complejidad baja.

Este submódulo estará a su vez diferenciado entre *servicios internos* y *servicios externos*.

- **Servicios Internos:** Aquellos servicios que no requieren de la coordinación con servicios fuera del entorno o que por su sencillez pueden ser resueltos de manera autónoma.
- **Servicios Externos:** Aquellos servicios que requieren de elementos o el uso de servicios adicionales y que no pueden ser resueltos localmente. Dado que estos servicios se implementarán dentro de la Estructura de Despliegue no podrán incluir mucha más funcionalidad que los internos, pero a diferencia de estos últimos, podrán solicitar los servicios a la capa de *Representación Simbólica Espacial*

### 3.6.5 Representación Simbólica Espacial

De manera análoga a la definición de la estructura de despliegue, en este submódulo se implementarán los servicios de alta complejidad.

Una característica diferenciadora de los servicios definidos en este submódulo debe ser la independencia del entorno particular y de la implementación concreta, manteniendo un nivel máximo de abstracción.

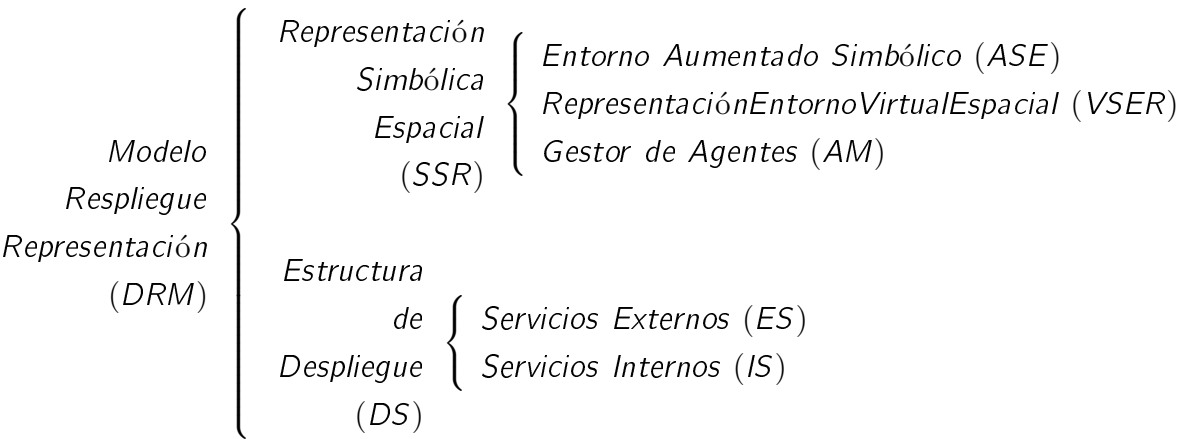
Los servicios que se implementen a este nivel serán capaces de interactuar con la *estructura de despliegue* a través de los *Servicios Externos* identificados, pero también podrán ofrecer un interfaz de usuario genérico para su monitorización y manejo.

Este submódulo se dividirá a su vez en tres componentes:

- **Representación Entorno Virtual Espacial:** O «**VSER**» en sus siglas en inglés. Si tenemos en cuenta que todo este modelo está pensado única y exclusivamente para entornos espaciales tridimensionales, es imprescindible que exista una representación simplificada del mismo. Como se ha comentado en la definición del  $ch\beta$  en el punto 3.5.6 toda la información que el módulo de *representación y despliegue* es capaz de manejar es una simplificación del total de información del entorno  $\Omega$ . Este es el componente que se encargará de gestionar esta información.
- **Entorno Aumentado Simbólico:** O «**ASE**» en sus siglas en inglés. El adjetivo aumentado se usa en el nombre de este módulo para remarcar su naturaleza de *cubrir* todas las necesidades de información que el entorno por sí mismo no puede hacer. Al igual que en la realidad aumentada se cruza información espacial existente para sobreimpresionar una imagen real, el conocimiento de este módulo servirán a los agentes para completar la representación virtual del entorno.
- **Gestor de Agentes:** O «**AM**» en sus siglas en inglés. Ya hemos definido este modelo como específico para la gestión de **S.M.A.S.** Debe por lo tanto tener un control de los agentes a los que esté dando servicio. Adicionalmente se implementarán los servicios necesarios e inherentes a los sistemas multiagente como páginas blancas o amarillas.

3.6.6 Esquema de componentes del modelo despliegue - representación

El modelo propuesto se representa de manera esquemática en función de los componentes identificados a nivel teórico:



### 3.7 Descripción de la metodología PIIVE-DM

En realidad no es necesario seguir una metodología para construir un sistema **V.R** ni de otro tipo. En múltiples ocasiones se ha conseguido construir sistemas exitosos únicamente siguiendo la intuición e incluso existe una corriente de pensamiento (Feyerabend, 1986) denominado «anarquismo epistemológico» que defiende la falta de método como método.

Si tomamos los ejemplos de usos de la **V.R** mostrados en el capítulo 2 podremos constatar que ninguna de las experiencias mostradas dispone de un método a seguir a la hora de construir los experimentos tratados. Para esto existen dos razones de peso. En primer lugar, dicha metodología no existe. Pero independientemente de este hecho existe otra razón: aunque en diferente medida todas las experiencias son categorizables como «pruebas de concepto». El objetivo de estas «pruebas de concepto» no es construir una versión inicial o temprana de un diseño de producción, sino únicamente demostrar que una nueva técnica es ciertamente posible. Este tipo de actividades es mayoritariamente existente en tecnologías nuevas o inmaduras, como es nuestro caso.

Pero existe una limitación a las «POC» que limita su aplicabilidad en disciplinas complejas. En concreto al estar trabajando con un sistema muy complejo en el que tenemos que usar distintas tecnologías hardware y software (renderers, motores de física, sistemas multiagente, gafas visualización estereoscópica, guantes hápticos. . . ), las pruebas de concepto se centran en probar una tecnología u otra, obviando el resto. Esto nos da una visión muy sesgada e incompleta de la validez del experimento. Por ejemplo, es habitual encontrar experimentos en los que se dedica mucha atención al hardware de visualización montando incluso instalaciones que abarcan habitaciones completas pero que dejan de lado la calidad del contenido representado. El resultado final para un neófito al observar el experimento suele ser bastante más pobre del real. Esta situación es muy habitual en investigación académica lo que por otro lado es lógico al estar íntimamente ligadas el entorno universitario y la investigación.

Pero habiendo dicho todo esto es necesario hacer frente a los problemas derivados de la falta de metodología:

- No es posible embarcarse en la construcción de un sistema complejo con garantía de éxito
- En caso de tener éxito no será posible repetir los pasos dados, corriendo el riesgo de perder la ventaja de la experiencia

- No existen garantías de que la solución hallada sea la mejor, sólo que es una de las existentes

Por último, existe una senda que parte de la artesanía y termina en la ciencia (Ebert, 1997) ganando madurez en el camino.

### 3.7.1 Preguntas frecuentes sobre esta metodología

- **¿Cómo se denomina esta metodología?**

El primer nombre que se le dio a esta metodología fue «Metodología de Simulación Paralela» aunque fue rebautizada más tarde a un nombre más descriptivo: «Metodología de Desarrollo Práctica Iterativa para Entornos Virtuales Inmersivos», también referida en ocasiones en inglés como «Practical Iterative Immersive Virtual Environment Development Metodology» y sus siglas **PIIVE-DM**.

- **¿Cuándo es aplicable esta metodología?**

Siempre que el objetivo sea construir un **Entorno Virtual Inmersivo** para ofrecer uno o varios servicios. En términos más generalistas, cuando queramos construir un entorno **V.R.**

## 3.8 Pasos de la Metodología

Los pasos metodológicos a seguir al construir un **V.I.E** no son muy distintos de los de construir cualquier otro sistema informático. Por esta razón en la presente metodología se han asimilado preceptos y mecanismos clásicos como UML de otros como Agile o SOA. También existe un interesante paralelismo entre el *diseño de sitios web* al encontrarnos con términos similares como *contenido usuarios* o *servicios* así como al diseño de videojuegos ya que en numerosas ocasiones un videojuego podría entenderse como un subtipo de **Entorno Virtual Inmersivo**.

Esta metodología se define como una serie de pasos a seguir de manera secuencial. Cada paso consta de una serie de subtarefas que puede ser realizada de manera secuencial o no y en general deben ser interpretadas según las peculiaridades del caso y usando el sentido común. Los cuatro pasos básicos son:

- Paso 1. Definición de Requisitos
- Paso 2. Arquitectura de Servicios
- Paso 3. Estudio de Despliegue
- Paso 4. Diseño de Tecnología

Esta metodología también se apoya en algunas herramientas de trabajo en forma de checklist que se definen en el punto 3.9

A continuación se definen los pasos de la metodología:

### 3.8.1 Paso 1. Definición de Requisitos

La lista de requisitos debe ser un texto que, aunque no tenga un formato rígido, si que es conveniente esté lo más estructurado posible. Si un requisito es de muy alto nivel conviene descomponerlo en subrequisitos. Deben utilizarse, en la medida de lo posible, frases cortas, concisas y asertivas. Se deben evitar también definiciones implícitas o vagas teniendo en cuenta que los puntos oscuros muchas veces ocultan funcionalidades no detectadas en primer término. Por último conviene tener en cuenta que las decisiones que se tomen en la definición de requisitos afectarán al sistema final por lo que es recomendable dedicarle tiempo suficiente a esta fase sin saltar con prisa a las siguientes.

Vamos a definir cuatro apartados dentro de la definición de requisitos, cada uno respondiendo a estas cuatro preguntas sobre el sistema que queremos desarrollar: ¿qué? ¿dónde? ¿quién? y ¿cómo?

- **¿Qué?** - Requisitos de Servicio
- **¿Dónde?** - Requisitos de Entorno
- **¿Quién?** - Requisitos de Público
- **¿Cómo?** - Requisitos de Despliegue

#### 3.8.1.1 Requisitos de Servicio ¿Qué?

Las cuestiones a realizar en esta parte comenzarán con la palabra «¿qué...» o redactado de otra manera similar pero siempre intentando describir el objeto principal del servicio.

**¿Qué servicios queremos ofrecer?** Estamos diseñando un servicio publicitario, el canal de acceso a un servicio de banca online, el auditorio de un servicio docente, ....

**¿Existe una manera de ofrecer el mismo servicio fuera de un VIE?** El canal de acceso al usuario de un servicio puede ser enormemente heterogéneo (web, apps móviles, chat, teléfono, kioskos, puntos informativos, ...) pero el servicio suele mantenerse constante a través de los canales, aunque con peculiaridades influenciadas por el canal. El conocer cómo se está dando este servicio en otros canales facilitará mucho el trabajo de análisis.

**En tal caso ¿ofrecemos algún valor añadido?** Aunque el servicio permanezca inalterado, cada canal aporta valores distintos: por ejemplo el canal telefónico ofrece facilidad de acceso en detrimento del resto, o un kiosko ofrece un interfaz muy cómodo en detrimento de la inmediatez.

Desde hace varios años, numerosos autores (Berman and Thelen, 2004; Dholakia et al., 2010; Otto and Chung, 2000) han definido la «multicanalidad» como concepto clave en la evolución de la oferta de servicios de las compañías a las nuevas generaciones. También apuntan a un principio muy sencillo que determina el éxito o el fracaso de las adaptaciones de los servicios a los nuevos canales: *si dados dos canales A y B con el mismo nivel de servicio, el canal A es más accesible que el canal B, el servicio ofrecido por el canal B tiene más riesgo de fracasar.*

Esta metodología está orientada a los **V.I.E**, lo que significa que para que un usuario pueda experimentar el entorno deberá tener acceso a un hardware que (hoy por hoy) no es ni cómodo ni fácil, es decir, la accesibilidad del «canal **V.R**» es baja. Aplicando el principio antes establecido es muy recomendable, en caso de que sea un servicio ofertable por otro canal, encontrar un valor añadido que compense la baja accesibilidad. En el caso de no encontrar ningún valor se debería justificar claramente la necesidad real de un **V.I.E**, o al menos replantear la manera de ofrecer el servicio.

**¿Qué interfaz de servicio se presentará a los usuarios?** El servicio o servicios se presentaran a los usuarios en cualquier forma que planteemos, pero esa implementación deberá ser llevada a cabo por uno o varios actores: seres humanos interactuando a

través de sus propios avatares, elementos del entorno dotados de IA, objetos virtuales de tecnología simple...?

Una vez tengamos expresados por escrito los requisitos de servicio tomaremos la taxonomía de servicios tal y como se define en 3.4.12 repasando uno a uno todos los servicios definidos en la taxonomía y comprobando si aplica o no aplica a nuestro sistema. Se recomienda usar las herramientas definidas en el punto 3.9.

### 3.8.1.2 Requisitos de Entorno ¿Dónde?

Ahora deberemos realizar cuestiones que permitan definir qué esperamos del entorno. El «entorno» es el concepto más diferenciador de esta metodología, además de ser un concepto clave de todo el estudio.

**¿Es nuestro entorno una réplica de otro entorno existente?** Es decir ¿vamos a «copiar» un entorno físico que ya existe? ¿vamos a generar uno desde cero? ¿será un modelo mixto? Existen numerosas opciones que podemos contemplar.

Un caso de uso típico en el que nos centraremos más adelante de un **V.I.E** son los «mundos espejo» o «simulación paralela» (Ángel Arroyo et al., 2010). Este tipo de instalaciones consisten en diseñar un entorno virtual como una réplica de un entorno real y disponer de la tecnología suficiente en ambos entornos como para replicar las acciones que ocurren en el real hacia el virtual y viceversa. En otros casos el entorno virtual también es una réplica de un entorno real, pero no existe ese «paralelismo», por ejemplo en las réplicas de entornos turísticos. En ambos casos podemos sustituir el entorno real por otro entorno virtual al que replicar.

También existen casos de uso en los cuáles el entorno virtual es totalmente original. En este caso se deberá definir en este punto las necesidades arquitectónicas del entorno y declarar todos los parámetros. Para estas tareas el analista deberá poder contar con la ayuda del arquitecto tal y como aparece en la sección 3.10.3

**Bocetos y planos** Aunque no es una pregunta, es una necesidad. Independientemente de si es un entorno original o replicado, es muy recomendable el uso de diagramas, bocetos, borradores de planos o cualquier otro material gráfico que permita a los artistas gráficos entender mejor cual es su tarea.



**¿Dónde tendrán los usuarios accesos a los servicios?** Siguiendo la idea de los bocetos, es recomendable el uso de storyboards para presentar las expectativas del caso de uso desde el punto de vista del usuario.

### 3.8.1.3 Requisitos de Público ¿Quién?

Para este punto es interesante utilizar la metodología «Personas» (Pruitt and Grudin, 2003) ideada por Alan Cooper (Cooper et al., 2004) y que consiste (de manera muy resumida) en diseñar durante esta fase una serie de usuarios concretos imaginarios arquetípicos y dotarles de toda la información que podamos para comprender mejor cómo va a interactuar con el sistema. Con este conjunto de usuarios arquetípicos podremos generar una matriz cruzando los servicios. Además de esta estrategia tendremos que respondernos algunas de estas preguntas:

**¿Qué tipo de público tendremos en nuestro servicio?** Será muy recomendable definir con todo el detalle posible qué público se espera tener en el entorno a análisis.

**¿Será multiusuario?** Un entorno multiusuario implica que los usuarios podrán acceder al servicio al mismo tiempo. Deberá dejarse constancia del comportamiento esperado cuando un servicio sea utilizado por más de un usuario ¿se permitirá? ¿hay un límite de simultaneidad? ¿cómo se define ese límite?

**¿Será un servicio abierto o cerrado al público general?** Los conceptos son totalmente transportables un entorno real: si un lugar está abierto al público general se han de observar ciertas precauciones de privacidad. Si es un servicio cerrado se deberán definir las condiciones de acceso.

**¿Cómo se gestionarán las credenciales de acceso?** Para acceder al V.I.E los usuarios normalmente harán uso de un avatar. Es necesario definir bajo qué credenciales un usuario podrá tomar control de un avatar y cómo se gestionarán las mismas.

**¿Qué permisos tendrán los usuarios?** Esta pregunta entra en relación con los servicios ofertados. Se parte de la base que no todos los usuarios tienen porqué tener acceso a todos los servicios.

**¿Con qué dispositivos podrán acceder los usuarios al VIE?** Aunque esta cuestión se escape un tanto del alcance de esta metodología es recomendable incluir toda la información posible que pueda afectar a la manera en la que los usuarios experimentan el entorno. Si para poder usar ciertos servicios es necesario disponer de cierto hardware, será recogido en estos requisitos.

#### 3.8.1.4 Requisitos de Despliegue ¿Cómo?

Por último se definirán todos los requisitos que no dependan de los servicios, usuarios o entornos, es decir todos los detalles adicionales que aunque no sean claves para el entorno, si es necesario tener en cuenta. Normalmente las restricciones, asunciones y consideraciones deben recogerse en este punto

**¿Qué horario tendrá este servicio?** 24x7, 16x5, 8x5, ...

**¿Será un servicio bajo demanda o siempre presente?** Si es bajo demanda se deberán especificar cuales serán los canales para solicitar el servicio y los detalles asociados a su solicitud como pago, duración, persistencia, acceso al público, etc.

**¿Existen requerimientos no funcionales como alta disponibilidad, capacidad de usuarios en paralelo, necesidad de trabajar en un sistema operativo determinado?** Y en general cualquier otro requisito que no sea clasificado en los capítulos anteriores. También se considerarán plazos para realizar el diseño e implementación del entorno, condiciones de aceptación, métricas de éxito, periodo de pruebas, periodo de garantía, soporte puesta en marcha, mantenimiento, etc.

### 3.8.2 Paso 2. Arquitectura de Servicios

Este segundo paso de la metodología se centra en los requisitos de servicio elaborados en el punto anterior. El resto de requisitos, aunque pueden ayudar a comprender mejor los servicios, serán estudiados en el tercer paso de la metodología.

**VIEs con alto índice inmersivo y ecológico** En este punto lo principal es ver de toda la arquitectura posible si realmente necesitamos toda y si realmente vamos a construir un **V.I.E** con alto índice inmersivo y ecológico. Recordemos que el modelo **DRM** está específicamente diseñado para este tipo de entornos. Otro tipo de entornos menos inmersivos y/o ecológicos también pueden ser descritos mediante una simplificación de este modelo, pero no se aprovecha la potencia del mismo.

**Hoja de Servicios** Para ayudarnos en esta tarea se propone el uso de la siguiente «hoja de servicios» 3.9. En dicha hoja se encuentran relacionados todos los servicios identificados según la taxonomía del modelo **DRM**. Basta con repasar todos y cada uno de los servicios, indicando si el sistema que queremos construir requiere de esos servicios o no.

**Correspondencia Servicio a Módulo DRM** Todos los servicios de la taxonomía de servicios están relacionados con un módulo del modelo **DRM**. Esto significa que podríamos encontrarnos con sistemas que no requiriesen un determinado módulo. Si esto ocurre es necesario replantearse la necesidad de utilizar este modelo en toda su amplitud.

**Servicios Tecnológicos** Los servicios tecnológicos son, por definición, incompletos. Es muy probable que se deban añadir más servicios tecnológicos al final de la lista.

Al final de esta segunda fase deberemos tener claro qué servicios vamos a implementar y qué módulos vamos a requerir

### 3.8.3 Paso 3. Estudio de Despliegue

Una vez se han identificado todos los servicios que vamos a necesitar es el momento de asociarlos al esquema de módulos **DRM** para poder acotar la estructura de despliegue. En este paso también se dará cabida dentro del despliegue de todos los requisitos «no servicios» identificados en el primer paso.

Al final de este paso, y una vez se tiene claro que módulos del **DRM** vamos a necesitar implementar, es el momento de decidir con qué software vamos a hacerlo.

**Módulos DRM especializados** Para poder decidir los detalles más técnicos de la despliegue software definido es imprescindible referirnos a los requisitos de entorno, públicos y sobre todo de despliegue definidos en el primer paso

Dependiendo de todos estos requisitos se deberán tomar un conjunto de decisiones. Esta metodología no pretende forzar la utilización de un producto software en detrimento de otro por lo que proponemos un estudio general del entorno concreto en el momento de la construcción del sistema. Al día de la redacción de la presente tesis no existe un claro *estándar de industria* y en general la tecnología es inmadura, incompleta y muy cambiante.

Ninguna metodología puede dar todas las respuestas de manera prefabricada, pero sí que es posible simplificar el problema y proponer buenas prácticas. En este sentido definimos esta relación de correspondencia entre servicios y módulos **DRM** que se define en la figura 3.8 en la sección 3.9.5.

Los módulos del modelo **DRM** ya están orientados al encapsulamiento de servicios por lo que podremos particularizar tres de ellos por especialización: ODM, PEM y RCM.

**«ODM» Object Data Module** También citado en la definición teórica como  $\Omega$ . Este módulo es el encargado de dar soporte de datos a todo el sistema. El único servicio identificado que recae en este módulo es el de persistencia, lo que significa que todos los objetos y estados del sistema son gestionados por este módulo.

- Como es obvio, este módulo debe implementarse a través de un gestor de base de datos. Existen numerosas soluciones de base de datos en el mercado, tanto de pago como OpenSource, y deberá elegirse el que mejor convenga en cada caso.
- Algunos ejemplos de gestores de base de datos utilizados en experimentos reales son: MySQL, Postgres o SQLite. Todos estos sistemas caen dentro de la categoría de bases de datos relacionales, pero no existe una razón para no utilizar otro tipo de soporte de datos.
- También es posible desarrollar un módulo de base de datos específico para el sistema en construcción aunque dado lo específico del servicio requerido y la enorme amplitud de software especializado que existe, no parece una decisión muy sensata a no ser que exista una razón de diseño de gran peso que la respalde.

**«PEM» Physical Engine Module** Todas las interacciones no inteligentes entre objetos deben ser realizados por este módulo, tal y como se ha definido claramente incluso a nivel teórico. Aunque sea un comentario trivial, es fundamental discernir entre los motores físicos y la representación gráfica de un entorno virtual. Como base sirva decir que los motores físicos no tienen por qué tener ningún tipo de representación visual.

- Igualmente existe una gran cantidad de productos software «physical engine» e incluso existe hardware dedicado (conocido como procesadores vectoriales) y especializado en los cálculos matemáticos requeridos para la ejecución de estos motores.
- Como ya se ha comentado anteriormente, a mayor calidad en el motor físico, mayor índice ecológico con lo que es fundamental en este punto buscar una solución que sea capaz de dotar al sistema de la verosimilitud que se requiere
- Como se definió en el punto 3.5.6 dentro del modelo teórico, el «PEM» tiene un canal de tipo  $\alpha$  ( $ch_\alpha$ ). Esto significa que el volumen de información que debe trasegar entre el «ODM» y el «PEM» es muy alto. Este detalle debe ser tenido en cuenta a la hora de definir la arquitectura de comunicaciones intra-módulos y asignar a este canal un ancho de banda adecuado. Una buena estrategia para conseguir este objetivo es minimizar la distancia física en el despliegue entre el «ODM» y el «PEM». De esta manera será menos costoso tener un ancho de banda suficientemente amplio como para albergar el  $ch_\alpha$ .

**«RCM» Renderer Controller Module** Este módulo es el encargado de implementar lo que habitualmente denominamos «GUI» por las siglas de «Graphical User Interface» aunque en nuestro caso no es el término más apropiado. En un sistema complejo como el que estamos diseñando deberán existir distintos puntos de interfaz con distintos roles de usuarios y por lo tanto el término es equívoco. Esta es la razón por la que el módulo se denomina «Renderer Controller», al desempeñar una labor dual.

- **Renderer:** Un «renderer» es un término muy común en la disciplina de los gráficos generados por computadora «CGI». Hace referencia a un tipo de software que, dado una serie de datos de naturaleza geométrica y de cualidades materiales, es capaz de generar una imagen en 2D similar a la que se produce dentro de una

cámara o un ojo humano. En el caso de que nuestro sistema deba contemplar la posibilidad de presentar una experiencia inmersiva, la existencia de un *renderer* es absolutamente indispensable. Dentro del contexto de la presente metodología, el componente que denominamos «*redenderer*» podría mostrar sonido e información háptica al usuario, por lo que no debemos quedarnos únicamente con la acepción visual del concepto. Como ya se ha comentado anteriormente, los «CGI» son un campo de investigación puntero y en pleno desarrollo por lo que siempre que se lleve a cabo un estudio de idoneidad de la tecnología a utilizar.

- **Controller:** El usuario, además de recibir la información visual, auditiva y háptica, deberá poder intervenir en el entorno. Estas intervenciones serán transformaciones directas en la información del entorno. Si un usuario mueve un objeto la información de ese objeto deberá ser modificada en el «ODM» siempre que se cumplan las reglas adecuadas. Al igual que en el «*renderer*» las posibilidades de realizar acciones por parte del usuario serán muy dependientes del hardware disponible.
- El canal que se utiliza es  $ch_\gamma$  tal y como se ha propuesto en la definición teórica, este canal requerirá de un ancho de banda reducido. Esto nos permitirá, al contrario que con el «PEM», separar la localización física del «RCM» y el «ODM» y reducir el ancho de banda necesario.

**Proveedores de Servicio y de Entorno** Denominaremos *Proveedor de Servicio* a aquel producto software en forma de librería, plataforma, servicio integrado o paquete de terceras partes que podremos utilizar en el despliegue de nuestro entorno transfiriéndoles la responsabilidad de la ejecución de ciertos servicios. El objetivo es aligerar la complejidad del sistema, así como hacer abordable el tiempo de implementación.

Una vez se hayan estudiado las distintas posibilidades es necesario decidir qué módulos van a ser implementados a través de *proveedores de servicios* y qué módulos serán desarrollados. Este paso no es sencillo y seguramente sea necesario realizar algunos ensayos y pruebas de concepto para ser conscientes de la complejidad de las plataformas existentes y su rendimiento. En la sección 3.10 se amplía la información sobre proveedores de servicio y de entorno y se dan algunos ejemplos.

### 3.8.4 Paso 4. Diseño de Tecnología

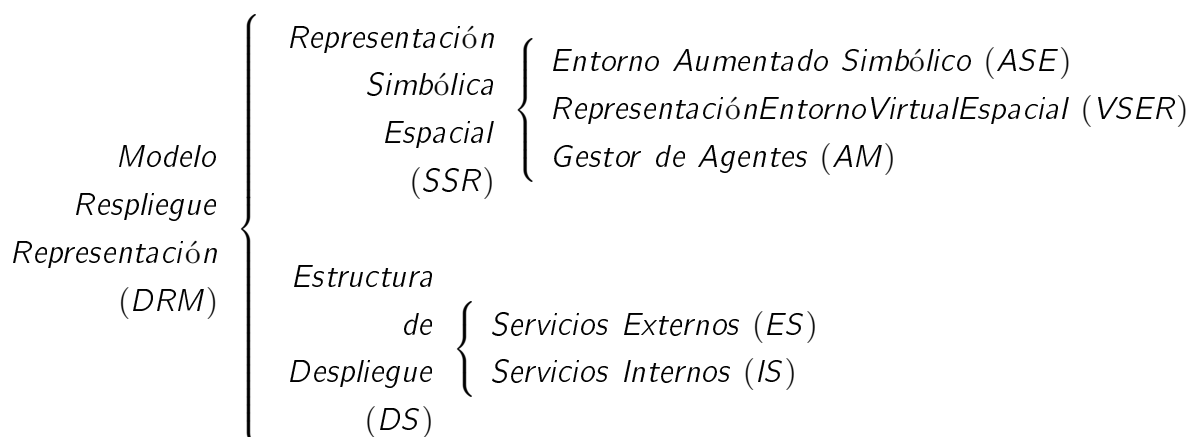
En este cuarto y último paso se diseñará la tecnología del entorno a estudio. Como en el paso anterior ya se ha definido qué *proveedores de servicio* y/o *proveedores de entorno* se usarán, en este punto encontraremos dos actividades primarias: definir cómo diseñaremos los servicios que no están cubiertos y preparar la integración con los *proveedores de servicio* y/o *proveedores de entorno*.

**Desarrollo de Servicios Tecnológicos** Habitualmente los servicios que más fácilmente pueden ser delegados en los *proveedores de servicio* son los ecológicos ( $\Psi_{eco}$ ) ya que son los más estandarizados. Sin embargo los servicios tecnológicos ( $\Psi_{tec}$ ) son habitualmente diferentes en cada implementación y dependen en gran medida del uso final del entorno.

Por esta razón en este paso el estudio del entorno se centra en los servicios tecnológicos ( $\Psi_{tec}$ ) y en definir las estrategias que puedan resultar más beneficiosas y óptimas para el desarrollo.

Como ya se ha definido en el capítulo 3.6 el módulo encargado de implementar estos servicios tecnológicos ( $\Psi_{tec}$ ) es el denominado «Módulo Despliegue-Representación». El nombre de dicho módulo responde a su naturaleza dual ya que se divide a su vez en dos submódulos: la «Estructura de Despliegue» y la «Representación Simbólica Espacial».

A su vez ambos submódulos se dividen en distintos componentes hasta quedar organizados de la siguiente forma:



En este paso será necesario revisar si la complejidad de los servicios tecnológicos a implementar requiere de la *Representación Simbólica Espacial* o abreviando según sus

siglas en inglés **SSR**.

Los **IS** (Servicios Internos) del componente **DS** (Estructura de Despliegue) pueden en algunos casos ser suficiente para implementar servicios tecnológicos suficientemente sencillos. Si se ha optado por usar *proveedores de entorno* que ofrezcan scripting, cosa bastante habitual, será necesario evaluar si la capacidad de cálculo y almacenamiento es suficiente o no. De ser suficientes los **IS** no será necesario realizar un análisis posterior.

En caso contrario, es decir, cuando los *servicios tecnológicos* sean complejos, requieran sincronización o simplemente el scripting del que dispongamos no sea suficientemente potente se deberá analizar en profundidad las necesidades y la estrategia de desarrollo. Para apoyar este paso se recomienda seguir la sección 3.11.

**Integración con los Proveedores de Servicio o Entorno** El último paso de la metodología es "poner todo junto". Llegados a este punto podemos habernos seguido una de las tres siguientes estrategias:

- No utilizar ningún proveedor de entorno ni de servicio, es decir, se desarrolla todo el sistema desde cero. En este caso, el más complejo, la integración se da por sentada en el propio análisis integral del sistema.
- Utilizar proveedores de servicio. En este caso de complejidad intermedia el trabajo de integración es alto y se debe prestar especial cuidado a la compatibilidad de modelo de datos entre distintos módulos. En este caso el **SSM** (Módulo de Soporte de Sistema) deberá encargarse de la transformación de datos adecuadamente para que cada submódulo sea compatible.
- Utilizar un proveedor de entorno. Este es el caso más sencillo ya que el proveedor de entorno ya es operativo y por lo tanto está integrado. Sin embargo se deberá diseñar el componente **DS** y especialmente los **IS** de manera adecuada y compatible con el proveedor designado.



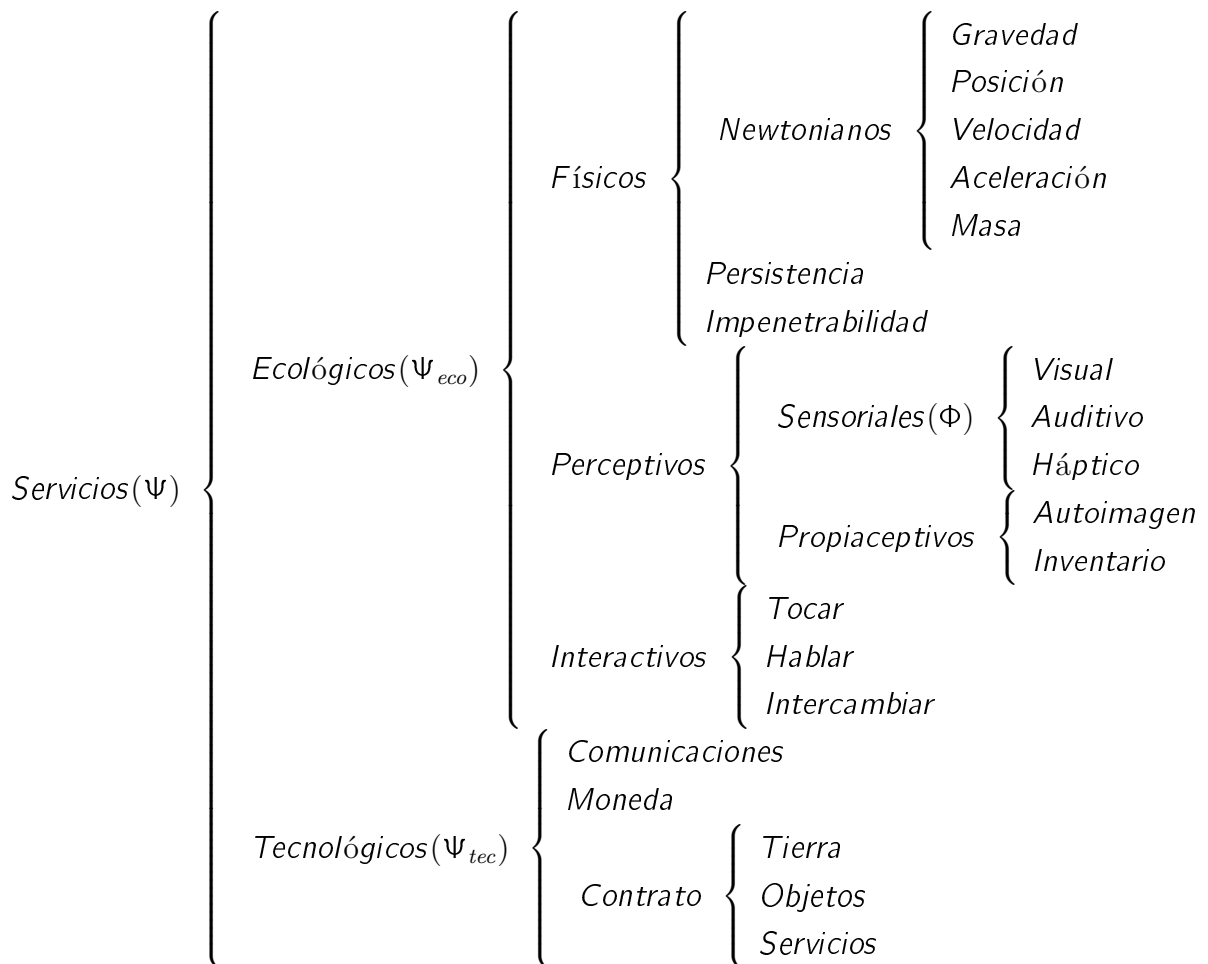
### 3.9.1 Hoja de Servicios

[illegible]

Figura 3.5: Hola de Servicios

### 3.9.2 Esquema Taxonomía completa de Servicios

A continuación se muestra el esquema taxonómico completo de servicios del modelo **DRM**



### 3.9.3 Diagrama de Módulos DRM

A continuación se muestra el diagrama de módulos del modelo teórico Despliegue-Representación según se definió en el capítulo 3.6

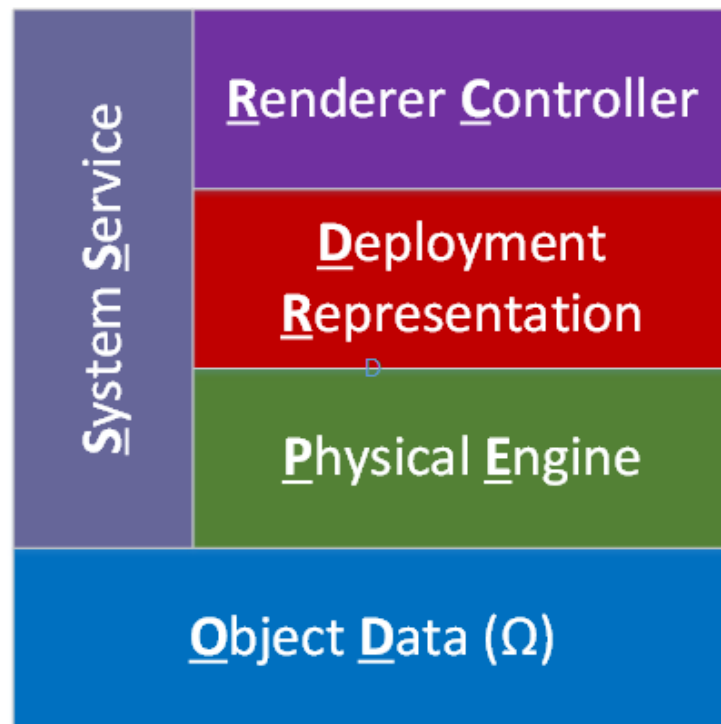


Figura 3.6: Diagrama de Módulos DRM

### 3.9.4 Correspondencia Taxonomía y DRM

A continuación se muestra la relación de correspondencia entre los servicios existentes en la taxonomía completa según se ha definido en el punto 3.4.12.

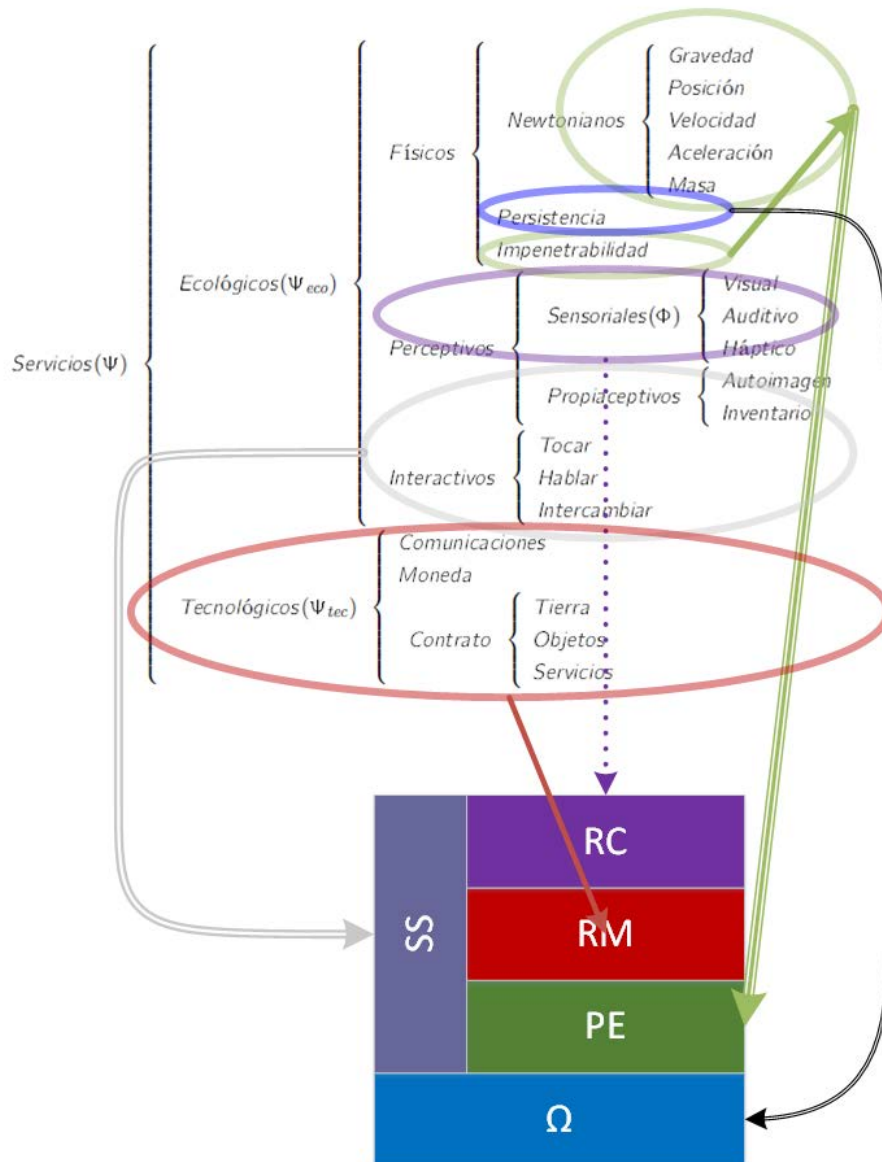


Figura 3.7: Correspondencia explícita entre la Taxonomía de Servicios y el Diagrama de Módulos DRM

### 3.9.5 Diagrama simplificado DRM con Servicios

El presente gráfico muestra el diagrama simplificado de Módulos **DRM** asociado a la Taxonomía de Servicios.

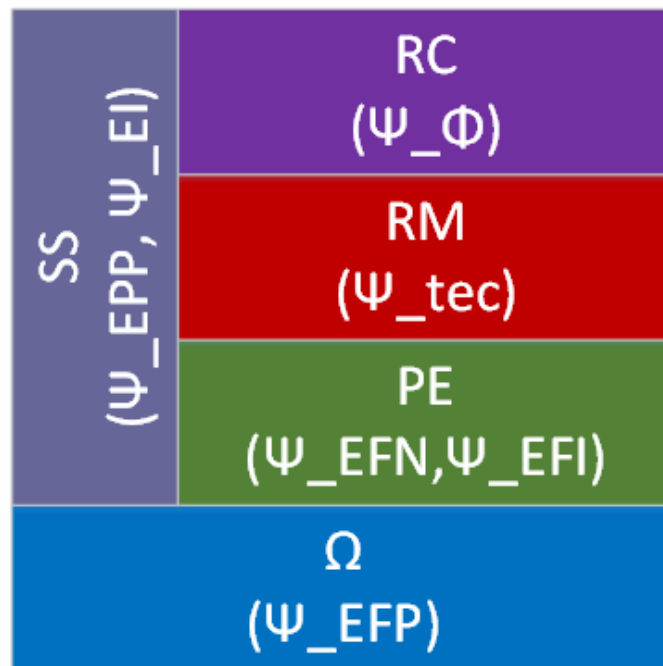


Figura 3.8: Diagrama Simplificado DRM con Servicios

## 3.10 Proveedores de Servicio y Proveedores de Entorno

En el paso 3 de la metodología (3.8.3) se define la necesidad de tomar la decisión de implementar los servicios a través de su desarrollo o bien buscar software de terceros que pueda operar como *Proveedor de Servicio*. Recordemos que se ha definido *Proveedor de Servicio* como: *aquel producto software en forma de librería, plataforma, servicio integrado o paquete de terceras partes que podremos utilizar en el despliegue de nuestro entorno transfiriéndoles la responsabilidad de la ejecución de ciertos servicios*.

A continuación se proponen algunas de las opciones actuales.

### 3.10.1 Motores de Juegos como proveedores de servicio

Un «Game Engine» o «Motor de Videojuego» es un tipo de software que agrupa una serie de técnicas de programación que permiten el diseño, la creación y la representación de un videojuego. Los motores pueden integrar algunas de estas funcionalidades:

- motor de renderizado 2D
- motor de renderizado 3D
- motor físico
- detector de colisiones
- sonidos
- scripting
- animación
- inteligencia artificial
- redes
- streaming
- administración de memoria

- escenario gráfico
- threading
- soporte regional
- cinemáticas

No es casual que muchas de estas funcionalidades se correspondan con los mismos servicios que se han identificado durante el análisis inductivo taxonómico del capítulo 3.4. Muchos de los juegos, sobre todo los más modernos, pueden ser definidos como un **V.I.E** o al menos como un entorno virtual.

El uso de *game engines* está muy extendido en la comunidad de desarrolladores de videojuegos por las claras ventajas que ello conlleva. Algunos de los motores más extendidos son: Bullet, Ogre, APE, Torque, Newton-Dynamics, Ogre, ioquake, Cry Engine, Unreal Engine o Unity.

En (Cowan and Kapralos, 2017) se comenta el término «serious games engine» como aquellos motores de videojuego cuya madurez y sofisticación los hacen utilizables para entornos «serios» o al menos que no persigan únicamente un objetivo lúdico. En este sentido podemos citar **Unity** como una de las opciones más utilizada en los últimos años.

Es por lo tanto viable utilizar alguno de estos «engines» como *proveedores de servicio* a conveniencia del *analista de servicios*.

### 3.10.2 Metaversos como proveedores de entorno

Aunque en algunas clasificaciones estas plataformas son categorizadas como «game engines» tradicionales desde el punto de vista de la metodología propuesta existen diferencias sustanciales que los hacen más interesantes para nuestro propósito.

Denominaremos *Proveedor de Entorno* a aquellos *Proveedores de Servicio* que ofrezcan un framework facilitador completamente funcional y operativo, siendo a la vez flexibles y programables.

Los «game engines» no pueden ser *proveedores de entorno* ya que no ofrecen un framework completo y funcional. En otras palabras, el «engine» por sí mismo no es capaz de ser experimentado. Por el contrario los proveedores de entorno ofrecen los mismos

servicios que los «engine» pero son completos y funcionales pudiendo ser experimentados y permitiendo que se construyan entornos de manera flexible.

La literatura denomina a estos *proveedores de entorno* de distintas maneras como *serious games* (Gros, 2016; Michael and Chen, 2005), *virtual worlds* (Alewine et al., 2016) o *metaversos* (Arroyo et al., 2011, 2009; Choi and Kim, 2017; Ángel Arroyo et al., 2010) aunque no existe, todavía, un criterio válido y unificado.

Los *proveedores de entorno* más conocidos son, a día de hoy:

- SecondLife <http://secondlife.com/>
- OSGrid <https://www.osgrid.org/>
- ViViWo <http://viviwo.virtway.com/>
- Active Worlds <https://www.activeworlds.com/web/index.php>
- Twinity <http://www.twinity.com>
- Anyland <http://anyland.com/>
- AltspaceVR <https://altvr.com/>

Los mayores handicaps que han experimentado estas iniciativas (ya sean con o sin ánimo de lucro) para conseguir una masa crítica de usuarios son:

- la dependencia del sistema final de la plataforma
- las dificultades de licenciamiento para que los usuarios finales creen su propio contenido.

Cabe destacar que OSGrid está implementado con la plataforma OpenSource **Open Simulator** [http://opensimulator.org/wiki/Main\\_Page](http://opensimulator.org/wiki/Main_Page) que es un fork del código fuente con el que se implementó SecondLife. Esto significa que **Open Simulator** puede ser utilizado de manera autónoma y ejecutar un *proveedor de servicio* personal sin coste asociado. Haciendo un paralelismo con la tecnología web, **Open Simulator** sería el «Apache» para entornos virtuales. Esta característica ha hecho que, como se verá en el siguiente capítulo, haya sido escogida en gran parte de los experimentos que se han llevado a cabo.



### 3.10.3 Roles implicados en la metodología

La metodología propone, inspirándose en los métodos ágiles y en concreto en SCRUM [https://es.wikipedia.org/wiki/Scrum\\_\(desarrollo\\_de\\_software\)](https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software)) los siguientes roles y sus responsabilidades generales:

- **Product Owner:** Mantiene la perspectiva de negocio, asegura la viabilidad económica del proyecto y marca el contenido del backlog.
- **Services Analyst:** Experto en la taxonomía **DRM**, debe ser capaz de analizar los requerimientos que genera el **Product Owner** y traducirlos en la lógica de negocio adecuada. Su labor principal es comprender la mejor manera de implementar los servicios requeridos.
- **Virtual Architect:** Un arquitecto virtual es un arquitecto de edificaciones especializado en las características especiales de construcción en entornos 3D. Será el responsable de que los escenarios y entornos creados en el **V.I.E** sean correctos y verosímiles.
- **Graphical 3D Artist:** El artista gráfico en 3D será el responsable, junto con el **Virtual Architect** de la imagen final del entorno.
- **Environment Provider Expert:** Dado que gran parte del trabajo consiste en integrar los servicios con el proveedor de entorno elegido, este experto será el responsable de conocer todos los detalles del mismo. También deberá asesorar al **Product Owner** y al **Services Analyst** sobre cómo utilizar los servicios del entorno de manera óptima.
- **Technical Developer:** Un ingeniero de software que desarrollará todos los servicios tanto en el **DS** como en el **SSR**.
- **Tester/QA:** El perfil tester debe ser capaz de probar el producto final y evaluar su calidad, realimentando el backlog.

### 3.10.4 Ejemplos de uso de VIEs

Existe un sinfín de posibilidades y ejemplos en la literatura tanto académica como de negocio que ya han sido recogidos a lo largo del capítulo 2. Aquí se listan sólo unos pocos

casos:

- Servicios Publicitarios
- Banca online
- Auditorio Servicio Docente
- Instalaciones Artísticas
- Réplicas de entornos turísticos
- Tienda de objetos físicos
- Entorno de Entrenamiento
- Entorno de Terapia

## 3.11 Metodología PIIVE para Servicios Tecnológicos complejos. PIIVE-STC

### 3.11.1 Campo de aplicación

En esta sección se describe la metodología a seguir en el caso concreto de **V.I.E's** con una necesidad de implementar *servicios tecnológicos* que no puedan ser llevados a cabo mediante lenguaje de scripting o herramientas similares.

Más concretamente en la sección 3.5.1 donde se describe el *experimento del comportamiento complejo* se define que el *comportamiento inteligente* necesario únicamente puede provenir de dos fuentes: inteligencia natural (**IN**) o inteligencia artificial (**IA**).

Para dar un ejemplo que ilustre este concepto aceptemos estos requisitos: «*en el diseño de un entorno virtual inmersivo destinado a terapia del comportamiento es necesario realizar un seguimiento de los avatares que interactúan en el mismo y anotar tiempos de respuesta ante estímulos. También es necesario tener un control sobre la terapia tal que si el paciente deja de atender, un monitor le preguntará verbalmente para reconducir la terapia*»

Podríamos abordar el problema sólo con **IA**, sólo con **IN** o con una estrategia mixta **IA+IN**. Veamos cómo se distribuiría la responsabilidad de la implementación de ese servicio en los tres escenarios:

- **Sólo IN**: Deberá existir un equipo de doctores o auxiliares visualizando en todo momento las evoluciones de los pacientes en el entorno y apuntando sus acciones. En el caso de que un paciente pierda atención lo podrán observar directamente y tomarán el control de un avatar para reconducir.
- **Sólo IA**: Una serie de agentes sensores estarán monitorizando constantemente la evolución de los pacientes y registrando en una base de datos un log de todas los parámetros de la terapia. Mediante técnicas de **IA** como segmentación, reconocimiento de patrones o cualquier otro, será capaz de detectar que un paciente ha dejado el curso de la experiencia. En ese caso se enviarán mensajes al paciente indicando que debe volver a prestar atención.
- **IA+IN**: Los agentes sensores **IA** monitorizarán y registrarán todos los parámetros con mayor detalle de lo que los doctores podrían hacer. Cuando un paciente deja de

responder según los parámetros de participación establecidos, será detectado por la **IA** y se lanzará una alerta a los doctores y auxiliares de la terapia para que puedan tomar el control de un avatar y dirigirse al paciente para reconducir la experiencia.

Obviamente, en este ejemplo el caso óptimo es la estrategia mixta. No siempre estará tan claro, pero por lo general las estrategias mixtas suelen dar mejor resultado que las puras. El *Analista de Servicios* deberá analizar cada servicio en detalle y elegir la mejor opción teniendo en cuenta todos los factores. En el ejemplo anterior parecía clara la opción mixta, pero si añadimos el *requisito de despliegue*: «este entorno deberá estar operativo en un horario ininterrumpido 24x7 y accesible a todo el público» puede que la opción *Sólo IA* no parezca tan descabellada.

La complejidad de los servicios a implementar también debe ser tomada en cuenta para definir con qué estrategias vamos a implementarlas. Los servicios sencillos, al igual que en el citado *experimento del comportamiento complejo* no tienen por qué ser implementados usando **IA**s o **IN**s. Los típicos servicios que no requieren el uso de técnicas avanzadas ni algoritmos complejos son los que recrean automatismos del mundo real: puertas automáticas, escaleras mecánicas, etc. En general, aparatos del entorno real de baja tecnología, que no requieran de memoria o capacidad de cálculo importante.

Una vez más, el *Analista de Servicios* deberá definir en cada caso la mejor manera de implementar cada servicio. A continuación nos centraremos en los servicios más complejos: los implementados a través de **IA**.

### 3.11.2 Implementación de Servicios mediante IA

En el verano de 1956 un grupo de insignes matemáticos, ingenieros, políticos y psicólogos se reunieron en el Dartmouth College (McCarthy et al., 1955) para una conferencia que marcaría el inicio y el propio nombre de la Inteligencia Artificial. Allen Newell y Herbert A. Simon, participantes de dicha reunión, estaban interesados sobre todo en el campo del *Human Problem Solving* (Newell et al., 1958). En aquella primera época, antes del invierno de la IA [https://en.wikipedia.org/wiki/AI\\_winter](https://en.wikipedia.org/wiki/AI_winter) los científicos pensaban que las máquinas podrían simular la capacidad humana para resolver problemas en los siguientes 10 años. Aunque como sabemos esas estimaciones estaban muy equivocadas, Newell y Simon presentaron durante aquel verano dos aplicaciones denominadas *General Problem Solver (GPS)* (Newell and Shaw, 1959) y *Logic Theory*

*Machine* (Simon, 1956) que programadas un ancestro del lenguaje LISP (IPL) eran capaces incluso de demostrar algunos de los teoremas del Principia Mathematica o resolver problemas como las *Torres de Hanoi*.

Trece años después de esta conferencia, Newell y Simon revisaban (Newell et al., 1972) el «Elements» (Newell et al., 1958) con mayor perspectiva. En dicho trabajo aplicaron la madurez adquirida durante los años de experimentación definiendo una estrategia basada en el procesamiento de la información: extracción de los procesos básicos de la tarea (problem solving) a realizar, generación de un «lenguaje» apto para procesar la información de dicha tarea, descomponer el problema en tareas sencillas dentro del «lenguaje» definido, probar, contrastar y refinar dicho proceso. Visto desde otra perspectiva (Russell et al., 2003) fueron capaces de implementar el algoritmo sugerido por Aristóteles 2300 años antes en su *Ética Nicomáquea* Libro III, en el que describe la voluntad humana y la deseable y óptima conexión lógica entre el pensamiento y las acciones.

Más de sesenta años después del bautizo de esta rama de la computación seguimos sin tener una idea clara de qué es la *inteligencia artificial* o incluso si este nombre es realmente adecuado. En cualquier caso vamos usar este término globalmente aceptado como un compendio de disciplinas, algoritmos, metodologías y técnicas computacionales aplicadas a la resolución de problemas generales.

Una de estas disciplinas, tal y como se ha comentado en el capítulo 2 dentro de la sección 2.3 es la denominada «**Sistemas Multi Agente Situados**» (**S.M.A.S** abreviado) que es especialmente interesante dentro del campo del estudio de la presente tesis.

Como ya se ha recalcado numerosas veces, esta tesis aborda el problema de la construcción de **Entornos Virtuales Inmersivos** mediante la «metáfora del entorno». Siguiendo este principio, los **S.M.A.S** son el enfoque idóneo para dotar de comportamiento inteligente los servicios tecnológicos que se requiera implementar en un **V.I.E.**

La sección (3.12) describe con el máximo detalle la metodología a seguir y propone una arquitectura específica para la implementación de servicios tecnológicos en **Entornos Virtuales Inmersivos** mediante **Sistema Multi Agente Situado**.

### 3.12 Sistemas Multiagente Situados en Entornos Virtuales Inmersivos

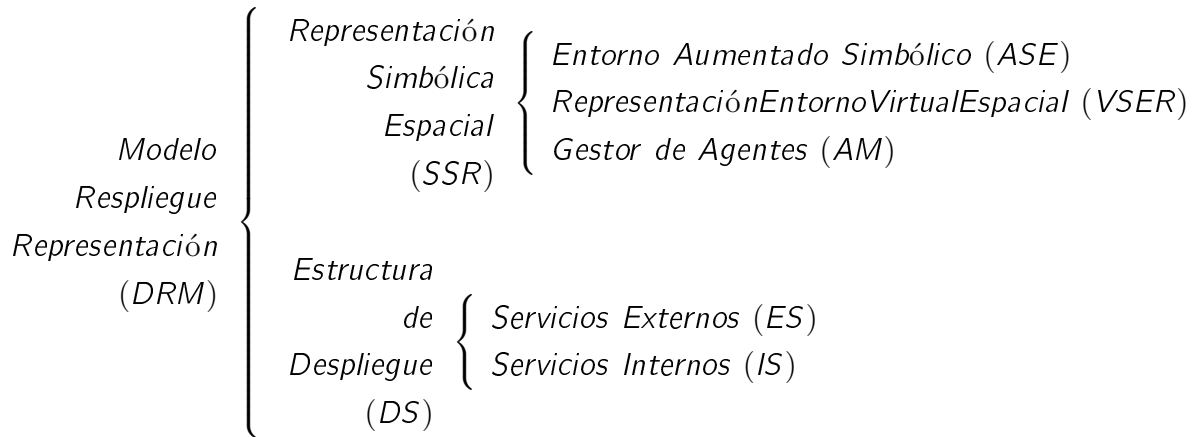
En esta sección se recopilan y unifican conceptos que se han ido definiendo a lo largo de todo el modelo teórico y de la metodología, dándole la forma de arquitectura que proponemos para el problema que da título a la misma.

El modelo y arquitectura que se propone está basado en el trabajo de diferentes autores expertos en la materia y en la propia experiencia del autor y directores de la presente tesis desde hace más de diez años.

Se debe enfatizar que tanto este modelo como la arquitectura subsiguiente es únicamente aplicable a los **Sistema Multi Agente Situado** en un **Entorno Virtual Inmersivo** con un contexto espacio-temporal. Este es el caso de la mayoría los agentes robóticos colaborativos (ya sean reales o virtuales) y otras muchas simulaciones en las cuales el espacio y sus dimensiones anchura, altura, profundidad y tiempo son la esencia del problema.

El modelo teórico que denominamos **Modelo Despliegue-Representación** ha sido definido en profundidad desde un punto de vista abstracto en el punto 3.6 por lo que en esta sección vamos a centrarnos en la aplicación práctica de este modelo teórico, matizando las distintas opciones de implementación y finalmente proponiendo una arquitectura en forma de *framework*.

Tal y como se define en el modelo teórico, es necesario diferenciar las dos partes del **DRM** y comprender la razón de esta división. Recordemos de nuevo el esquema de componentes teóricos del modelo:



**La estructura de despliegue (DS)** es la parte del **S.M.A.S** que está más ligada al nivel bajo, es decir que estaría en contacto con el *proveedor de entorno*. Aunque desde un punto de vista totalmente pragmático no sería necesario encapsular e independizar todo el desarrollo de los agentes del proveedor que escojamos para implementar el sistema, esa decisión iría en contra de las premisas de portabilidad y flexibilidad que hemos promulgado como principios de trabajo. Por otra parte el uso de un «middleware» es una buena práctica en ingeniería informática para cualquier tipo de arquitectura multicapa. Esta es la razón principal para definir la **DS** como un módulo bicomponente: un componente (el **IS**) interno que es dependiente del *proveedor de servicio o de entorno* concreto que estemos utilizando, y otro componente (el **ES**) que ofrecerá un interfaz común traducido hacia la capa superior (el componente **SSR**).

**La representación simbólica espacial (SSR)** es el componente que, abstraído del *proveedor de entorno o de servicio* agrupa, ejecuta y almacena toda la información de los servicios tecnológicos ( $\Psi_{tec}$ ) a implementar en el **V.I.E**. En la definición teórica se dividió este módulo en tres componentes: **ASE**, **VSER** y **AM** encargados respectivamente de la información simbólica del entorno, el almacenamiento de los datos espaciales y la gestión de los propios agentes. Como se verá más adelante, en función de la implementación podrán surgir algunos otros módulos adicionales dedicados a la gestión del sistema. Una instancia real de una implementación de un **SSR** será denominado **HUB**. Un **HUB** podrá dar servicio a un **DS** en un solo *proveedor de entorno* o a varios **DS** implementados en distintos *proveedores de entorno*, tal y como se apuntaba en la definición teórica en 3.4. Cuando el segundo caso ocurre tendremos el efecto denominado «Simulación Paralela».

**Dualidad de los Agentes en DRM** Esta división entre **DS** y **SSR** también se traslada a los agentes que se implementan bajo esta metodología. Todos y cada uno de los agentes que se diseñen siguiendo este modelo tendrán una parte localizada en el **DS** y otra en el **SSR**, aunque lo que podrá variar enormemente es la cantidad de carga funcional que se encuentra en uno o en otro módulo. Aunque este concepto se explicará en profundidad más adelante junto con el resto de circunstancias que lo hacen más comprensible, podemos adelantarlo en este momento haciendo un paralelismo entre la dicotomía existente entre los agentes *deliberativos* y *reactivos* (Corchado and Molina, 2002).

En la práctica nos hemos encontrado con que en los sistemas multiagente desarrollados con estrategia *reactiva*, los agentes tenían mucha más carga funcional en la **DS**, concretamente en los servicios internos (**IS**) que en **SSR**. Por el contrario, en los agentes *deliberativos* los agentes en el entorno eran meros canales de acceso a **IS** y que toda la carga de proceso se realizaba en **SSR**.

### 3.12.1 Estructura de Despliegue

Los cimientos del modelo y de la arquitectura propuesta se asientan en la *Estructura de Despliegue* (**DS**). Como Ferber y Muller (Ferber and Müller, 1996) anunciaron ya en 1996, el entorno es una pieza fundamental en la construcción de un sistema multiagente. Se definió entonces un **S.M.A.S** *Multi Agent System Situated* o *Sistema Multi Agente Situado*) como un sistema multiagente en el cual el entorno es una entidad de primer orden a la hora de definir sus características y construir el sistema.

En consecuencia hemos de prestar al entorno tanta atención como a los agentes. Por ejemplo, el modelo (FIPA, 2002) se basa en el “agent platform” que ha de tener un directorio facilitador (yellow pages), un sistema de gestión de agentes (white pages) y un sistema de intercambio de mensajes. Aunque por supuesto es factible implementar dichos servicios a través de los propios agentes, lo cierto es que son casos especiales que desvirtúan en cierta manera la uniformidad de la solución.

Weyns y otros autores en (Weyns and Holvoet, 2004; Weyns et al., 2005, 2007b) defienden a ultranza la necesidad de enmarcar dichos servicios «especiales» dentro del entorno, y con este fin proponen un interesante modelo en (Weyns, 2010). Para la construcción modelo propuesto se han tomado algunas de las características más útiles del modelo práctico de Weyns, aunque otros elementos han sido descartados.



Las ideas propuestas por Weyns que han inspirado el modelo práctico son:

- *La coordinación de los objetivos del sistema queda bajo la responsabilidad del entorno* (en nuestro modelo los objetivos se analizan bajo distintas perspectivas, una de ellas es el entorno)
- *La responsabilidad de hacer cumplir las «normas del juego» queda bajo la responsabilidad también del entorno. Por ejemplo, las reglas físicas impuestas por la naturaleza (gravedad) o reglas específicas cognitivas.* (en nuestro modelo se han refinado estas ideas en forma de servicios ecológicos)
- Niveles de soporte. Se plantean 3 niveles de soporte:
  - *Nivel básico* (en el presente modelo es equivalente a los **IS** del **DS**)
  - *Nivel de abstracción* (especialmente útil en agentes deliberativos para, por ejemplo definir los estados Markovianos del entorno) (sería equivalente a una versión rudimentaria del **SSR** mezclado con los **ES**)
  - *Nivel de interacción* (sobre todo en cuanto a la comunicación simbólica *agent2agent* y *agent2environment*. (en el modelo propuesto esta interacción simbólica espacial se lleva a cabo dentro del componente **ASE** del **SSR**)
  - *Existe un cuarto nivel denominado «de reflexión» en el que los agentes descargan la información espacial, aunque no está completamente aceptado por el resto de autores* (según el modelo propuesto sería algo parecido al componente **VSER** del **SSR**)

Un concepto que no existe en los trabajos de Weyns (ni en ningún otro hallado hasta la fecha) es el de *Simulación Paralela* y la estrategia de separar **DS** y **SSR** para conseguir la independización del proveedor de entorno.

La estructura de despliegue (**DS**) se asemeja al «nivel básico» de los niveles de soporte de Weyns, pero amplía y complementa este concepto. A través de la estructura de despliegue nos es posible sentar las bases de todo el sistema multiagente para uno o varios servicios concretos. Como ya se ha especificado en la metodología en cuatro pasos en el capítulo 3.8, previamente a la construcción o el diseño de los agentes en la estructura de despliegue hemos de definir las características del problema, definir los servicios que hemos de implementar, qué casos de uso tendremos que afrontar o los posibles usuarios

que nos encontraremos. Como en cualquier otro sistema informático, es extremadamente incauto comenzar a desarrollar antes de haber analizado lo que queremos hacer.

También se habrán de definir las métricas y métodos para evaluar el rendimiento del sistema, tanto por motivos informativos como para el posible uso de funciones de fitness. Estas métricas, desde el punto de vista teórico del modelo, no tienen cuerpo funcional, pero al implementar las soluciones la práctica ha demostrado que es importante tenerlas en cuenta y plantearlas en función de los requisitos de despliegue que hemos comentado en el punto 3.8.1.4. Por ejemplo, si un requisito de despliegue es algo tan empírico como que el consumo de CPU del motor físico no supere cierto porcentaje, deberemos definir la manera en la que nuestro sistema sea capaz de averiguar cuánto CPU está consumiendo dicho módulo. Este conjunto de servicios estará implementado dentro de los servicios internos (**IS**) y le daremos el nombre de «métricas de éxito».

Por lo tanto podemos afirmar que la estructura de despliegue está formada por: uno o varios *proveedores de entorno*, un conjunto de *servicios internos*, un conjunto de *servicios externos* y un conjunto de *métricas de éxito*.

### 3.12.1.1 Servicios Internos

Los servicios internos dependerán de cómo el sistema sea capaz de intercambiar información con el resto de módulos. Recordando lo definido en el modelo teórico en el punto 3.5.6, los servicios internos (**IS**) y por ende el módulo **DRM** intercambiarán información con el **ODM** ( $\Psi_{\Omega}$ ) a través del canal  $ch_{\beta}$ .

En la práctica esta interacción se llevará a cabo mediante un **API**(*application programming interface*), un **SPI**(*service provider interface*) o incluso un **EPI**(*environment provider interface*) y será significativamente distinto en cada caso.

Veamos un ejemplo de cada tipo a la hora de implementar un servicio muy común: el desplazamiento de un objeto en el espacio. Lo denominaremos  $\psi_{desp}$

- **API**: La información de los objetos del entorno está disponible en una base de datos SQL. Existe una tabla donde se encuentran todos los objetos existentes. En dicha tabla existen tres campos (x,y,z) donde se define su posición espacial. Cuando queramos ejecutar el servicio  $\psi_{desp}$  deberemos implementar una petición SQL de tipo **UPDATE** que cambie los campos x, y, z del objeto en cuestión. Tanto la

posibilidad como las consecuencias de este cambio deberán ser gestionadas por el **IS** (colisiones, interacciones, permisos, etc.)

- **SPI**: En este caso la información está encapsulada por un servicio al que accedemos en función de lo que su documentación disponga. Habitualmente el servicio lo estará ofreciendo un motor físico que tendrá un servicio  $\Psi_{\phi, desp}$  asociado a  $\psi_{desp}$ . En este caso solo habrá que «traducir» la llamada. El motor físico resolverá la petición en función de las interacciones con otros objetos físicos, aunque las interacciones con otros elementos de comportamiento complejo como avatares u agentes quedará sobre la responsabilidad del **IS**.
- **EPI**: Si contamos con un *proveedor de entorno* el proceso de llamada al servicio no se realizará directamente al motor físico  $\Psi_{\phi}$  sino al módulo de soporte de sistema **SSM** y será tratado como un servicio de interacción  $\Psi_{E.I.}$ . De esta manera el **IS** no tendrá la responsabilidad de propagar el cambio en todo el entorno, haciendo mucho más sencilla la implementación.

Como es lógico y ha quedado gráficamente demostrado, los servicios internos (**IS**) dependerán de la arquitectura que se decida en cada caso. Incluso dos sistemas que utilicen el mismo nivel de interfaz (**EPI** por ejemplo) y por lo tanto dos *proveedores de entorno* diferentes podrán tener distintos servicios internos.

### 3.12.1.2 Servicios Externos

Los servicios externos (**ES**) son los encargados de «estandarizar» de cara al exterior los servicios internos (**IS**) anteriormente citados. Además se encargarán de resolver las comunicaciones que sean necesarias en función de la arquitectura de red definida en cada caso.

Aunque la metodología que presentamos no tiene como objetivo definir todas las opciones posibles ni acaparar el total de soluciones existente, sí que pretende ser una guía de implementación y por esa razón vamos a proponer los siguientes servicios externos (**IS**) como una base sobre la que construir y extender en las posibles implementaciones. Obviamente, existe un fuerte paralelismo entre estos servicios que vamos a proponer y la taxonomía de servicios que fue presentada en el punto 3.4.12.

Los servicios externos siguen la arquitectura clásica cliente-servidor, pero pueden ser ofrecidos en tres perspectivas distintas:

- **Cliente *ES*, Servidor *AM* /  $ES \Rightarrow AM$**  : Bajo esta perspectiva, el agente está implementado como un *IS* que requiere de los servicios del *SSR* bajo demanda. Es el modelo de comportamiento típico de los agentes reactivos, partiendo de un *proveedor de entorno* con un alto nivel de scripting en el que los agentes tienen mayor entidad en el entorno virtual que en el *HUB*. Estos agentes serán los denominados **Metabots** que son definidos en el experimento que se describe en 4.2. En adelante y para abreviar, a los servicios ofrecidos bajo esta perspectiva los denominaremos «servicios reactivos».
- **Cliente *AM*, Servidor *ES* /  $AM \Rightarrow ES$**  : En la perspectiva complementaria, los agentes están implementados en el *SSR*, dentro del *gestor de agentes (AM)* y solicitan los servicios al *DS*. Esta es la perspectiva clásica para implementar agentes deliberativos, dando nombre también a los servicios correspondientes que serán clasificados como «servicios deliberativos».
- **Bidireccional /  $ES \Leftrightarrow AM$**  : Como en casi todos los entornos complejos en los que la realidad se circunscribe a una posición intermedia entre dos puntos, siempre existirán casos mixtos. En el modelo propuesto la implementación final del **S.M.A.S** dependerá de los requisitos de cada problema y del resto de circunstancias concretas, por lo que siempre será la mejor opción dejar el sistema abierto en ambos sentidos y no cerrar ninguna opción en el tiempo del diseño. La perspectiva bidireccional simplemente aúna las dos anteriores, permitiendo la existencia de agentes complejos que exhiban en un determinado momento características *deliberativas* y en otras ocasiones características *reactivas*.

### 3.12.1.3 Servicios Deliberativos Propuestos

Esta lista relaciona algunos de los servicios «mentales» que el modelo propone. Es decir, aquellos servicios que los agentes implementados en el *DS* solicitan al *SSR*.

- Gestión de Campos
- Pathfinding
- Planificación
- Segmentación
- Clusterización
- Scoring de objetos

### 3.12.1.4 Servicios Reactivos Propuestos

Esta lista relaciona algunos de los servicios «corporales» que el modelo propone. Es decir, aquellos servicios que los agentes implementados en el **SSR** solicitan al **DS**. Estos servicios son habitualmente los mismos que encontramos dentro de la taxonomía de servicios  $\Psi_{E.I.}$ , es decir los servicios ecológicos interactivos. Casi siempre están expresados como «órdenes» a ser ejecutadas por la parte reactiva del agente dentro del entorno.

- Aplicar impulso lineal a un objeto
- Aplicar momento giratorio a un objeto
- Desplazamiento a un punto en el espacio
- Tocar un objeto
- Teletransportar a un punto (remoto) en el espacio
- Emitir mensaje a un canal público (hablar/gritar/susurrar)
- Emitir mensaje a un canal privado (hablar/gritar/susurrar)
- Escuchar un canal publico
- Escuchar un canal privado
- Examinar un objeto
- Percibir los objetos (próximos/cercanos/lejanos)

### 3.12.2 Representación Simbólica Espacial

Revisando de nuevo los tres niveles de soporte que se proponen en (Weyns and Holvoet, 2004) y que ya hemos examinado en 3.12.1, el *nivel de abstracción* que describen permite dotar a los agentes deliberativos del suficiente nivel simbólico como para ser capaces de mantener un comportamiento complejo. Por otro lado, la componente social de un **S.M.A.S** situado es difícil de encajar únicamente con los recursos que nos ofrece el *proveedor de entorno*. En esta situación debemos obtener una «aumentación» del entorno de manera similar a como la «augmented reality» complementa la visión normal en la tecnología actual.

En el capítulo 2 ya se han presentado numerosas propuestas sobre sistemas multiagente en la que la lógica formal juega un papel fundamental. Desde que McCarthy (McCarthy and Hayes, 1969) en 1969 definió el «situation calculus» se ha utilizado en numerosas ocasiones como por ejemplo en (Reiter, 1978, 1991) o más recientemente por (Ferber and Müller, 1996). Aunque las técnicas se han ido refinando, el concepto original prevalece:

asunción de «mundo cerrado» y la expresión del estado actual del entorno y de las acciones de manera formal. Estas técnicas habitualmente requieren dos requisitos: gran capacidad de cálculo y representación simbólica del entorno y las capacidades de los agentes.

Ambas tareas por lo general requieren una enorme potencia computacional, así como una representación simbólica del entorno y técnicas que permitan a los agentes manejar este entorno. Las características minimalistas inherentes a los agentes en un **S.M.A.S** se oponen frontalmente a las necesidades computacionales de tamaño «mainframe» para ese objetivo. Además, la representación simbólica del entorno no debe ser realizada por el agente, sino por la estructura de soporte de nivel superior que se aprovecha de esta perspectiva de middleware para añadir la característica de intercambio de información entre los agentes.

El concepto práctico de *Representación simbólica espacial (SSR)* se define como un conjunto de **HUBs** distribuidos que proporcionan de forma colaborativa servicios de alto nivel a una *Estructura de Despliegue (DS)* y a todos sus posibles entornos subyacentes.

Otro concepto que vamos a utilizar es el de **HUB**. Esta palabra es muy utilizada en la tecnología y además con varios significados diferentes. Esta palabra significa en inglés eje, pero también «centro de actividad». Con ese sentido se ha denominado al conjunto de agentes que tienen una actividad común. También podría entenderse como el vecindario de agentes o el conjunto de los mismos. En este contexto, definimos **HUB** como el conjunto de servicios de alto nivel de abstracción sobre el que trabajan un grupo de agentes.

Los **HUBs**, además, deberán ser capaces de comunicarse entre ellos siguiendo un conjunto de reglas sociales y colaborativas establecidas. En algunas ocasiones dos o más **HUBs** pueden ser plenamente colaborativos hasta el nivel de fusionar automáticamente sus representaciones espaciales virtuales (permisos de acceso y modificación recíprocos), semi-colaborativos en los que sólo las representaciones e informaciones públicas de los **HUBs** serán consultados por el resto, pero sin posibilidad de modificarlo e incluso podrían existir **HUBs** competitivos que no comparten nada de información directamente. En este caso, sin embargo sí podrían estar sincronizados a través de un **HUB** neutral, por ejemplo, para tener información común y pública. Otra razón para separar en **HUBs** la representación simbólica espacial es la fuerte carga de procesamiento que suele estar asociada a los procesos cognitivos de nivel alto; de esta manera hacemos el

sistema escalable independientemente del número de agentes o del tamaño del espacio a representar.

Al igual que en el modelo teórico, la arquitectura práctica se divide en distintos componentes, unos teóricos (ya definidos con anterioridad) y otros prácticos:

- Teóricos, como se ha definido anteriormente en 3.6.5
  - **Representación Entorno Virtual Espacial:** O «**VSER**» en sus siglas en inglés. Si tenemos en cuenta que todo este modelo está pensado única y exclusivamente para entornos espaciales tridimensionales, es imprescindible que exista una representación simplificada del mismo. Como se ha comentado en la definición del  $ch\beta$  en el punto 3.5.6 toda la información que el módulo de *representación y despliegue* es capaz de manejar es una simplificación del total de información del entorno  $\Omega$ . Este es el componente que se encargará de gestionar esta información.
  - **Entorno Aumentado Simbólico:** O «**ASE**» en sus siglas en inglés. El adjetivo aumentado se usa en el nombre de este módulo para remarcar su naturaleza de *cubrir* todas las necesidades de información que el entorno por sí mismo no puede hacer. Al igual que en la realidad aumentada se cruza información espacial existente para sobreimpresionar una imagen real, el conocimiento de este módulo servirán a los agentes para completar la representación virtual del entorno.
  - **Gestor de Agentes:** O «**AM**» en sus siglas en inglés. Ya hemos definido este modelo como específico para la gestión de **S.M.A.S.** Debe por lo tanto tener un control de los agentes a los que esté dando servicio. Adicionalmente se implementarán los servicios necesarios e inherentes a los sistemas multiagente como páginas blancas o amarillas. *El **HUB** debe tener un control de los agentes a los que está dando servicio. En este módulo, además, es donde reside la funcionalidad que clásicamente se le atribuye al agente facilitador, a las yellow o white pages. Adicionalmente, en algunas representaciones, los agentes tienen distintos niveles de conocimiento: local y compartido con el **HUB** Cuando el conocimiento local es demasiado complejo o requiere demasiados recursos como para ser almacenado por el agente dentro del entorno, éste es el lugar donde se almacena, como una réplica de la representación virtual aumentada del espacio simbólico sólo para el agente.*

- Prácticos. Definidos a raíz de la experimentación y para dotar de capacidad operativa a la implementación del **HUB**

- **Soporte del Sistema:** O «**SS**». Aunque a primera vista pareciera que toda la información del entorno y del Hub queda almacenada en la representación del espacio simbólico, aun así existen datos que quedan fuera del dominio, por ejemplo la consolidación de datos sobre métricas de éxito. Dependiendo del problema estos datos pueden ser más o menos abundantes, con lo que el componente será más grande o más pequeño, pero el concepto sigue teniendo validez. Otro uso de este módulo es el contener el «Interfaz de Usuario» cuando este es poco relevante en el sistema.
- **Gestor de HUB** (parte del **VSER**). Este componente es únicamente necesario en el caso de que estemos implementando un sistema *multi-HUB*. Aquí reside la responsabilidad de hacer que el **HUB** provea los servicios apropiados, así como comunicarse con otros **HUBs** y establecer las relaciones apropiadas de confianza y rivalidad que el problema requiera. Este componente estará funcionalmente localizado como una parte clave del componente **VSER** para actuar como enrutador de mensajes entre los **HUBs**.
- **Interfaz de usuario** (parte del **SS**). Primero hemos de aclarar el término «usuario». Cuando en este apartado nos referimos al concepto «Interfaz de Usuario» no se refiere al usuario final del **V.I.E** que accederá al sistema mediante el módulo **RCM**, sino al *usuario* que está operando el **HUB**. Como en cualquier sistema informático deberá existir una manera de interactuar con el **HUB**: arrancar, parar, monitorizar, o cualquier otra acción que sea requerida. Por último el usuario también debe ser informado e incluso permitirle realizar parametrizaciones. Por ejemplo la información sobre las métricas de éxito debe ser siempre mostrada en este módulo.

Entendida esta puntualización, el usuario operador del **HUB** también debe estar informado e incluso se le debe permitir realizar cambios y parametrización en el **HUB**. Este sub-componente será implementado dentro del componente **SS** y normalmente toma la forma de una consola Web estándar con una interfaz gráfica que incluye la supervisión, personalización, parametrización y control de todo el sistema. Cabe señalar que, incluso no siendo un objetivo principal de un **S.M.A.S**, un gran desafío para el futuro y la popularización



de la tecnología es la falta de interfaces de usuario amigables. Si tenemos la intención de construir **S.M.A.S** de manera seria y profesional, se debe cuidar el aspecto exterior que percibirán los futuros operadores del sistema.

Para poder ilustrar cómo puede llevarse a cabo un **SSR** vamos a exponer y proponer una implementación real que ha servido como base a gran parte de los experimentos que se describirán en el capítulo 4. Esta implementación se denomina **SSR-HUB** y los detalles de los componentes se detallan en la sección 3.13.

En el siguiente diagrama se describe la arquitectura más compleja posible del sistema: El **DS** está compuesto por distintos *proveedores de entorno* y se conectan a un conjunto de **HUBs**. La estrategia de implementación de los agentes es mixta, es decir, existen *Agentes Deliberativos* en el **SSR** y *Agentes Reactivos* en el **DS**.

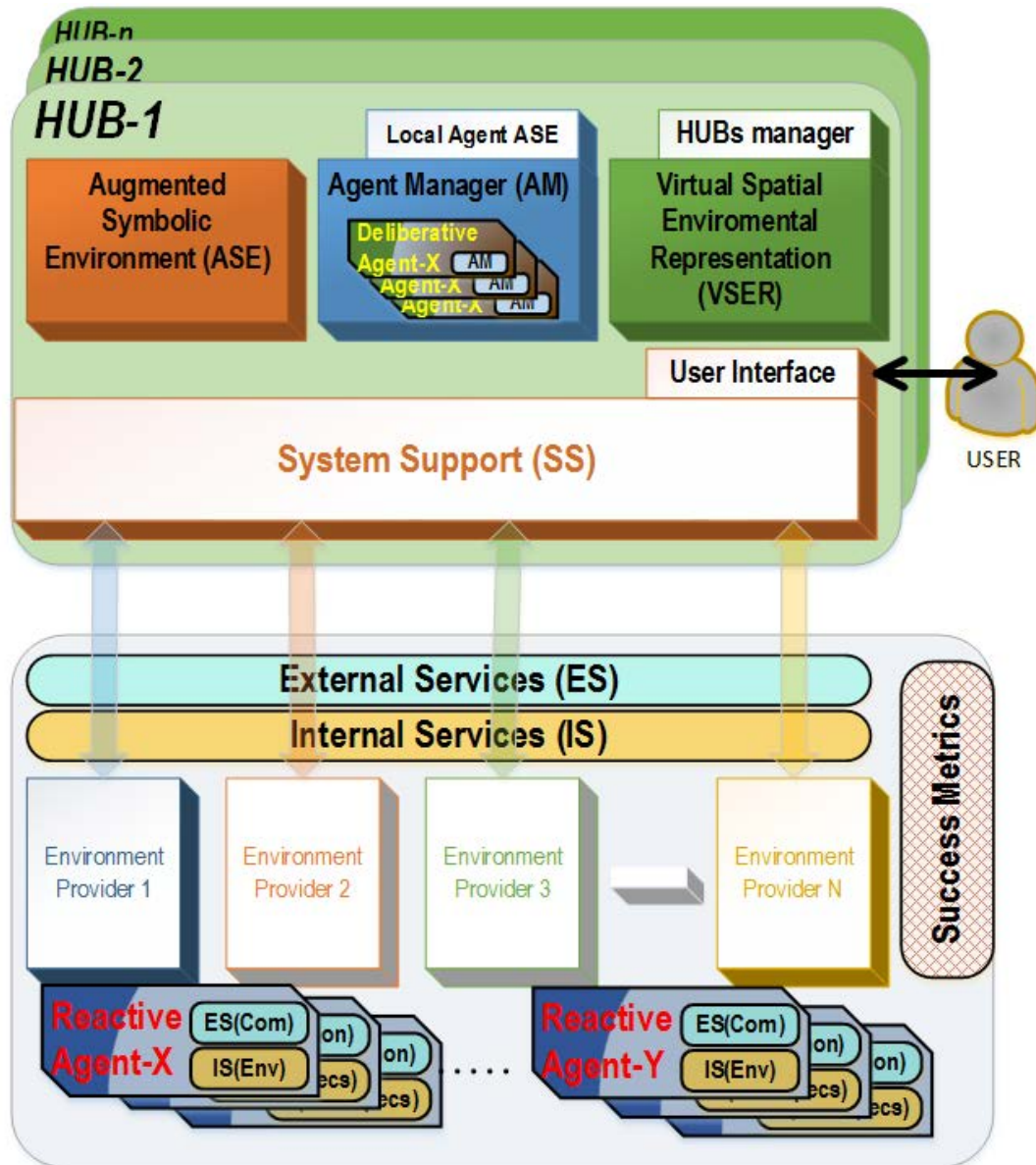


Figura 3.9: Arquitectura SSR Multi-HUB con Múltiples Proveedores de Entorno en Simulación Paralela

## 3.13 Framework SSR-Hub

### 3.13.1 Módulos software

El framework propuesto se ha desarrollado completamente en lenguaje JAVA mediante el IDE Eclipse. Se compone de los módulos principales:

- Representación Entorno Virtual Espacial **VSER**
- Entorno Aumentado Simbólico **ASE**
- Gestor de Agentes **AM**

y los siguientes módulos secundarios o de soporte:

- Estado del Sistema **SS**

### 3.13.2 Características técnicas de la herramienta

El framework se distribuye en forma de un solo archivo **JAR** ejecutable directamente a través de línea de comandos en cualquier sistema operativo para el cual se haya desarrollado una *JVM - Java Virtual Machine*. Es por tanto una aplicación multiplataforma.

Aparte de la necesidad de existencia de una *JVM* también requiere una conexión a un servidor de Base de Datos SQL en el cual almacenar toda la información del sistema. Esta conexión de datos es utilizada únicamente por el módulo **VSER** que es el encargado de la persistencia de datos del **HUB**.

Todas las conexiones provenientes desde los Agentes son recogidas y preprocesadas por el módulo **AM**, que posteriormente notifica al módulo **ASE** si tiene el servicio que está solicitando el agente y tiene permisos suficientes para utilizarlo. Además el módulo **ASE** consolida y aplica las reglas de fusión de datos que sean necesarias por el servicio.

A continuación se muestra un ejemplo de las opciones de ayuda y configuración de las que dispone la herramienta:

```
oscar@haven ~/apps/ssrhub $ java -jar ssrhub.jar -h
Deployment-Representation Model. Spatial Symbolic Representation Hub
```

Version v2.1.19 4/2/2017

```
-v --version  opcional sin argumentos
               <<version>>

-h --help     opcional sin argumentos
               <<help>>

-a --agentport opcional 1 argumento [port]
               <<Port to listen for Agents messages>>
               * Arg 1 [port] de tipo Number por defecto {XXXX}

-c --consoleport opcional 1 argumento [port]
               <<Port to listen for Hub Web Console>>
               * Arg 1 [port] de tipo Number por defecto {XXXX}

-p --prefs     OBLIGATORIO 1 argumento [file]
               <<Name of the file for preferences storage>>
               * Arg 1 [file] de tipo String

-x --adminpin  OBLIGATORIO 1 argumento [pin]
               <<Administration pin code>>
               * Arg 1 [pin] de tipo String

-u --userpin   OBLIGATORIO 1 argumento [pin]
               <<Users pin code>>
               * Arg 1 [pin] de tipo String

-db --database OBLIGATORIO 5 argumentos [server] [port] [scheme] [user] [pass]
               <<MySQL connection data example: 111.111.111.111 3306 scheme user password>>
               * Arg 1 [server] de tipo String
               * Arg 2 [port] de tipo Number
               * Arg 3 [scheme] de tipo String
               * Arg 4 [user] de tipo String
               * Arg 5 [pass] de tipo String

-el --externallanding opcional 1 argumento [page]
               <<External landing page>>
               * Arg 1 [page] de tipo String

-al --agentlimit opcional 1 argumento [limit]
               <<Agent Web Server limit petitions per second>>
```

```
    * Arg 1 [limit] de tipo Number por defecto {10}

-cl --consolelimit opcional 1 argumento [limit]
    <<Console Web Server limit petitions per second>>
    * Arg 1 [limit] de tipo Number por defecto {10}

oscar@haven ~/apps/ssrhub $
```

### 3.13.3 Interfaz gráfico de la herramienta

Aunque como se ha indicado la herramienta es lanzada desde línea de comandos con un interfaz CLI básico, dispone de un interfaz gráfico basado en una consola web. Para poder acceder a esta herramienta es únicamente necesario disponer de un navegador y acceder a la URL y el puerto que se ha indicado al **SSR-Hub** en la parametrización anterior.

Para hacer más usable esta consola web se ha diseñado pensando en el manejo desde dispositivos móviles, aunque es compatible con navegadores de escritorio. Estas son algunas muestras del interfaz gráfico:

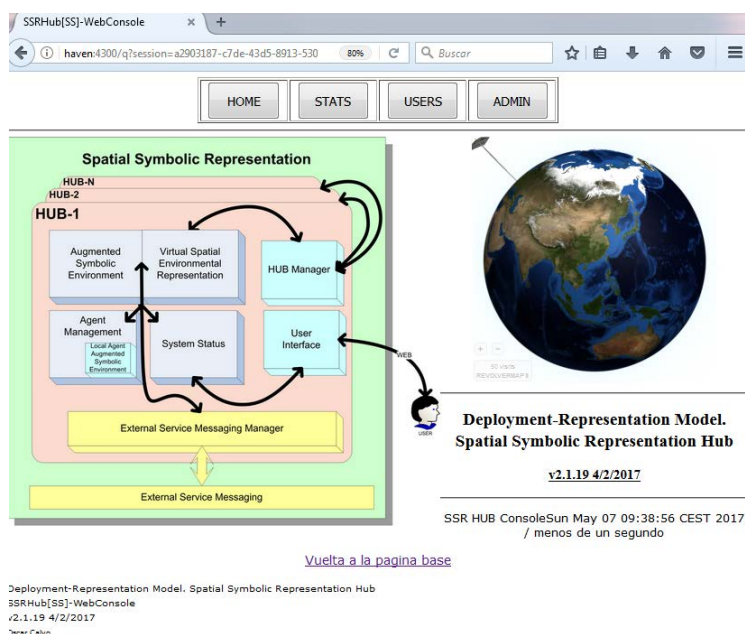


Figura 3.10: Vista HOME

Desde la vista «HOME» el **SSR-Hub** nos muestra la página de landing (luego es personalizable) y un widget con la localización geográfica de las últimas conexiones. En la parte superior se muestra la botonera de navegación que aparecerá en todas las pantallas y nos permitirá usar una opción u otra.

Mediante el botón «STATS» se muestra la información en tiempo real de la que dispone el módulo **ASE**.

Mediante el botón «ADMIN» se abre el submenú de administración

SSRHub[SS]-WebConsole

haven:4300/q?session=a2903187-c7de-43d5-8913-530

HOMESTATSUSERSADMIN

Module ASE (Augmented Symbolic Environment)

Current Agent View

AgentID	AgentName	Region	Owner Name	Owner Key	LK Location	LK Time	IP
0eec76a7-ec23-4cbb-bc1b-c4129cc020c8	MSO-agent-example	DRMLab Data Flock Experiment (1001, 1000)	oscar drmlab	ce6f	(122.858400, 142.130500, 21.176500)	9 segundos	192.168.1.1
a8629835-13f3-40e1-afcd-2c3bb5d9682a	MSO-agent-example	DRMLab Data Flock Experiment (1001, 1000)	oscar drmlab	ce6f	(125.579300, 146.319100, 21.176500)	menos de un segundo	192.168.1.1
ce2baa9c-54b7-40f5-bb39-7c8956a81e70	MSO-agent-example	DRMLab Data Flock Experiment (1001, 1000)	oscar drmlab	ce6f	(121.621800, 145.905700, 21.176500)	3 segundos	192.168.1.1
c585c706-20e0-46ed-b142-95c2c3794de8	MSO-agent-example	DRMLab Data Flock Experiment (1001, 1000)	oscar drmlab	ce6f	(122.475200, 149.141100, 21.176500)	5 segundos	192.168.1.1
ea5530ef-1e61-41f7-87cc-de6437164adb	MSO-agent-example	DRMLab Data Flock Experiment (1001, 1000)	oscar drmlab	ce6f	(126.068800, 149.340000, 21.176500)	7 segundos	192.168.1.1
227df622-4361-4905-9c30-7d96fa814df8	MSO-agent-example	DRMLab Data Flock Experiment (1001, 1000)	oscar drmlab	ce6f	(125.481100, 144.478900, 21.176500)	9 segundos	192.168.1.1
31a74a04-eae1-42e2-bbf7-d157dd313083	MSO-agent-example	DRMLab Data Flock Experiment (1001, 1000)	oscar drmlab	ce6f	(123.722200, 144.446500, 21.176500)	menos de un segundo	192.168.1.1
6d776f5a-1ae5-4618-bf81-8d25d157bf6a	MSO-agent-example	DRMLab Data Flock Experiment (1001, 1000)	oscar drmlab	ce6f	(125.095800, 151.304600, 21.176500)	3 segundos	192.168.1.1
a202a988-1b3d-461a-bf88-6657627bae7c	MSO-agent-example	DRMLab Data Flock Experiment (1001, 1000)	oscar drmlab	ce6f	(124.280400, 151.983900, 21.176500)	5 segundos	192.168.1.1
551bffff-26b8-41d6-bc59-ea7d20e970fc	MSO-agent-example	DRMLab Data Flock Experiment (1001, 1000)	oscar drmlab	ce6f	(127.409800, 147.945700, 21.350810)	59 segundos	192.168.1.1

Figura 3.11: Vista del botón STATS

SSRHub[SS]-WebConsole

haven:4300/q?session=a2903187-c7de-43d5-8913-530

HOMESTATSUSERSADMIN

STOPCONTROL PANELPREFSMONITOR

SSR HUB ConsoleSun May 07 09:57:55 CEST 2017 / menos de un segundo

[Vuelta a la pagina base](#)

Deployment-Representation Model. Spatial Symbolic Representation Hub  
SSRHub[SS]-WebConsole  
v2.1.1.19 4/2/2017  
Oscar Calvo

Figura 3.12: Vista del admin menu

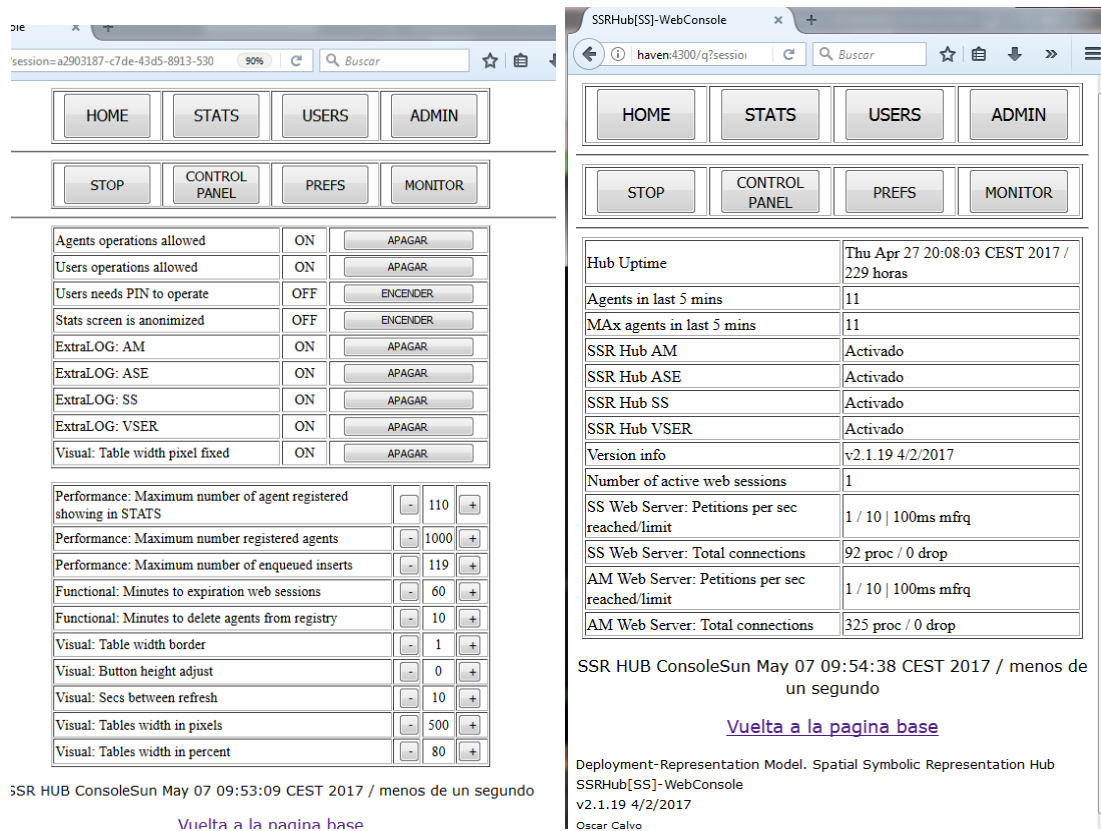
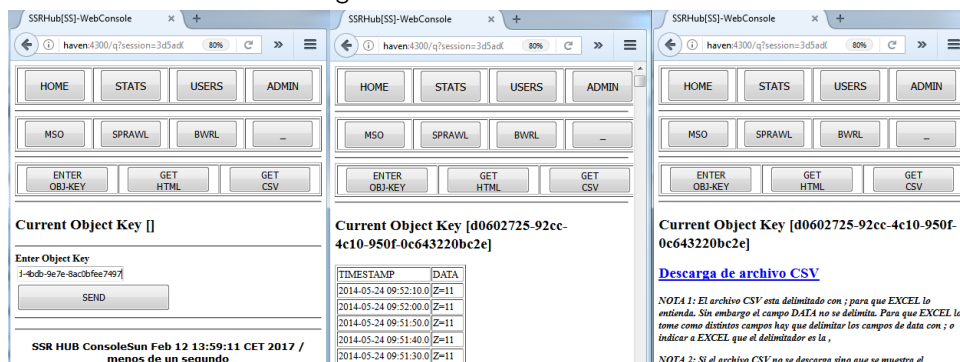


Figura 3.13: Vista de Preferencias y Panel de Control

En la opción ADMIN/PREFS se administran las preferencias del sistema. En la opción ADMIN/CTRL PANEL se monitoriza el panel de control, que posee información estadística de rendimiento en tiempo real.

Figura 3.14: Interfaz de Usuarios



Este es un ejemplo de interfaz web para los usuarios



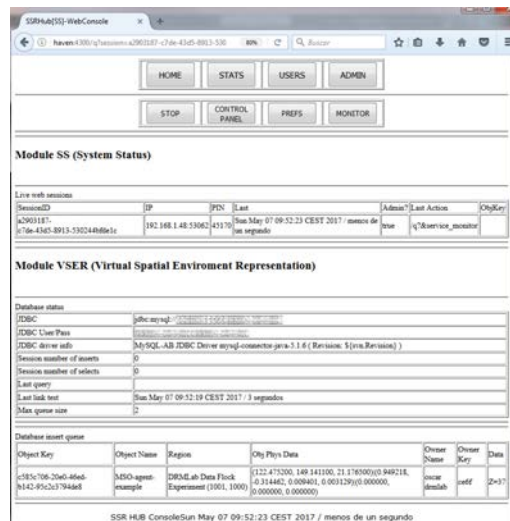


Figura 3.15: Vista del monitoring vser ss

En la opción ADMIN/MONITOR es posible monitorizar el estado de los módulos **VSER** y **SS**, es decir, tanto las conexiones web de la consola actual como las conexiones abiertas con el motor de la base de datos y la salud de la misma.

### 3.13.4 Seguridad de la herramienta

La naturaleza multiusuario y pública de la herramienta requiere un nivel de seguridad alto para mantener tanto la integridad de la herramienta como la seguridad de los datos de los usuarios. Estos son algunos ejemplos de medidas de seguridad incluidas como: introducción de pin, segmentación de permisos, control de usuarios, etc.



Figura 3.16: PIN / STOP / Not allowed

## 3.14 Conclusiones

En este capítulo se han descrito las **once** propuestas con las que cubrir los objetivos definidos en el punto 1.3.1. Posteriormente se ha definido la metáfora del Entorno, caracterizando su tecnología de soporte y algunos principios de diseño como la «Validez Ecológica», el «Principio de Máxima Verosimilitud» y el «Flujo de Compromiso de Presencia».

Después hemos realizado una formalización abstracta del entorno, definido como un conjunto topológico de estados, objetos y servicios. Para poder trabajar con estos conceptos se han propuesto algunas medidas de similitud entre Servicios como la «Función de similitud simple» o las «Condiciones de similitud entre conjuntos de servicios». También se han formalizado las funciones de similitud.

Posteriormente nos hemos centrado en definir algunos casos particulares de entorno como los euclídeos de grado 3 o el entorno real, con el objetivo de poder dar la definición matemática completa del «Índice de Inmersión» y el «Índice de Validez Ecológica».

También hemos definido una «taxonomía de Servicios», identificando servicios ecológicos físicos a partir de la observación del mundo real. Posteriormente hemos enunciado el modelo arquitectónico **DEC2RPM**, desglosado las distintas interacciones físicas complejas y formulado una taxonomía de interacciones inteligentes entre elementos según su naturaleza. Dentro del modelo **DEC2RPM** se describen los tres canales de interacción y los módulos: Motor Físico, Interprete - Controlador, Despliegue - Representación y Soporte de Sistema.

Posteriormente se describe el modelo Despliegue - Representación, que se divide entre Estructura de Despliegue y Representación Simbólica Espacial

También se ha descrito la metodología iterativa **PIIVE-DM**, definiendo sus pasos, que se dividen en Definición de Requisitos, Arquitectura de Servicios, Estudio de Despliegue y Diseño de Tecnología. Sin olvidar las herramientas de soporte a la metodología. Posteriormente se ha definido una metodología práctica para la implementación de Servicios Tecnológicos complejos, algunas notas sobre la implementación de Servicios mediante IA para terminar con una breve reseña sobre el «**Framework SSR-Hub**».

---

# 4

## Experimentación

*«¿Qué es real? ¿Cómo defines lo real? Si estás hablando de lo que puedes sentir, lo que puedes oler, lo que puedes saborear y ver, entonces lo real son simplemente señales eléctricas interpretadas por tu cerebro».*

- Morfeo

### 4.1 Introducción

En el presente capítulo vamos a repasar **cuatro** experiencias que dan validez a las propuestas que se han realizado en los capítulos anteriores. Dentro de cada experiencia se estudiará en detalle los objetivos concretos de las mismas, haciendo especial énfasis en identificar cuáles de los objetivos concretos que nos hemos marcado al comienzo de la presente tesis son realizados total o parcialmente por cada experiencia.

Algunas experiencias cuentan a su vez con uno o varios experimentos que caracterizan un objetivo principal. Los objetivos principales cubiertos son:

- Probar la validez de la arquitectura **DRM** propuesta
- Probar la validez de la metodología **PIIVE-DM** propuesta
- Probar la validez de los frameworks de trabajo **MetabotLib** y **SSR-Hub** propuestos

Las experiencias que describiremos son las siguientes:

- **SL/2009:** Con *MetabotLib* y «*AvatarRank*»
- **SL/2011:** Con *MetabotLib*, *SSR-Hub* y *Sprawl1*
- **OS/2012:** Con *SSR-Hub* y *Sprawl2*
- **OS/2016:** Con *SSR-Hub* y *MSO*

## 4.2 Experimento SL/2009

### 4.2.1 Introducción

Este es uno de los primeros experimentos que fueron llevados a cabo en la Isla TESIS dentro del **V.I.E Second Life** durante los años 2009 y 2010. El objetivo de dicho experimento era comprobar la **Hipótesis 1** *metáfora antropomórfica adecuada*.

La hipótesis refinada para este experimento fue expresada como sigue: *Dentro de un V.I.E en el que por definición es imposible diferenciar a un avatar manejado por un humano o por una IA, se plantea que: a mayor realismo gráfico e interactividad del avatar, mayor predisposición tendrán los usuarios a interactuar con dichos avatar-IA's*

Aprovechando el acceso público a la plataforma **Second Life** y la constante afluencia de usuarios ajenos al experimento se planteó el siguiente modelo:

- Crear tres **Metabots** con tres aspectos y comportamientos diferentes para disponer de tres niveles de realismo:
  - el **Usher** o «Ujier» poco realista
  - el **Pretender** o «Pretendiente» con un mayor realismo
  - el **Chatterbox** o «Parlanchín» con el mayor nivel de realismo posible
- Los tres **Metabots**, aunque tenían distinto comportamiento, eran capaces de almacenar toda la información de interacción: identificación de contertulio, charla, frases, tiempo entre frases, número de frases por charla, etc.
- Exponer a los tres **Metabots** en una zona concreta y recoger datos durante ocho semanas.
- Al final de la fase de recogida de información se estudiarían los resultados para valorar si la hipótesis de partida se cumplía o no.

El simple hecho de plantear este experimento implicaba ciertas necesidades tecnológicas que debieron ser abordadas, casi todas ellas orientadas a obtener algoritmos que fuesen capaces de interpretar las acciones humanas (¿cuándo empieza y termina una conversación? ¿cómo interpretar que un avatar está hablando a otro? ¿qué lenguaje no

corporal espera ver un humano en un avatar para que entienda que es otro humano?...)) y emular estas, sobre todo en lo relativo a la comunicación verbal.

Desde el punto de vista de la presente Tesis, este fue el primer experimento que se realizó, por lo cual tanto las herramientas teóricas como prácticas eran aún incipientes. Sin embargo, y a pesar de esta circunstancia, fue plenamente funcional para corroborar un gran número de las hipótesis planteadas en el punto 1.3.2.

En los siguientes secciones se describe la experiencia y las conclusiones obtenidas. Parte del contenido y los resultados de este experimento fue publicado en los artículos (Arroyo et al., 2009) y (Arroyo et al., 2011).

## 4.2.2 Desarrollos teóricos del experimento SL/2009

### 4.2.2.1 Definición de Metabot

El término *Metabot* es ampliamente usado en esta experimentación y requiere de una pequeña definición. De la manera más concisa posible podríamos definirlo como «*un Metabot es un elemento software capaz de manejar un avatar dentro de un V.I.E de una manera indistinta de como lo hace un ser humano*»

Entrando más en detalle, un *Metabot* es un robot software que se desenvuelve en un V.I.E. Un *Metabot* debe ser capaz de manejar uno o varios avatares para experimentar el V.I.E de la misma manera que los seres humanos. Los metabots pueden ser tan externos al V.I.E como lo son los seres humanos, e incluso el código que los conforma puede estar ejecutándose en otro sistema. Son un nuevo tipo de interfaz mucho más intuitivo a los humanos ya que han de estar contruidos siguiendo el principio de máxima verosimilitud con el mundo real y en particular con el antropomorfismo humano. De esta manera la interacción hombre-máquina es intuitiva y no requiere un aprendizaje especial. Desde un punto de vista humano un *Metabot* es otro avatar más que con el que interactuar conversacionalmente y desde el punto de vista de la máquina es un sistema multiagente basado en el conocimiento que utiliza indistintamente cualquier canal de datos como sistemas de recuperación de información, ontologías, bases de datos, motores de indexación, redes sociales, sistemas de recomendación, información pública, etc.

Un *Metabot* también es un agente móvil, proactivo y perceptivo. Debe ser capaz de moverse en un entorno espacial virtual inmersivo de manera similar a como lo hacen

los seres humanos que experimentan ese mismo entorno. También debe ser capaz de comunicarse con otros avatares, ya sean manejados por metabots o por humanos, si bien es cierto que con los humanos ha de utilizar preferentemente sistemas conversacionales. Podrá utilizar cualquier sistema experto, multiagente o ACL a la hora de representar el conocimiento necesario para llevar a cabo su funcionalidad. De igual manera para ser capaz de interactuar en entornos complejos tridimensionales persistentes deberá apoyarse en ontologías que construyan una base de conocimiento antropomórfico común con el de los seres humanos.

#### 4.2.2.2 AvatarRank

Uno de los primeros problemas que surgen cuando se implementan *Metabots* es definir los parámetros de la comunicación humana dentro de los **V.I.E's** y cómo construir *Metabots* sensibles a estos parámetros. Por ejemplo, ¿cómo puede un *Metabot* saber cuándo un avatar está interesado en *él/ella/ello*?

Los seres humanos llevamos millones de años de evolución para ayudarnos en estas tareas, y sin embargo a veces es difícil para nosotros saber cómo comportarse en actividades con alta carga social. En cualquier caso, los seres humanos instintivamente saben lo que es un círculo de conversación quién es la persona más popular en la sala.

Si seguimos la **metáfora del entorno**, estos comportamientos relacionados con la percepción humana del «**entorno**» deben ser transportados a los **V.I.E's** con la mayor fidelidad posible y respetando al máximo el **principio de máxima verosimilitud**. En los **V.I.E's** hay existen los mismos conceptos de cercanía, tacto, atención, etc.

Es por lo tanto indispensable para construir *Metabots* capaces de integrar estos comportamientos absolutamente antropomórficos aislar y definir ciertos algoritmos que nos ayuden a dar medidas coherentes de estos comportamientos.

En este sentido vamos a definir el «**AvatarRank**» y que definimos como: *«aquella función que permite evaluar dentro de un **V.I.E** cómo de **popular** es un avatar»*

Cuando usamos la palabra «**popular**» nos referimos a la cantidad de atención que está recibiendo un *avatar* en un momento dado. Por lo tanto, cuanto más alto es el rango de un *avatar*, más atención recibe de otros *avatares* (ya sea humano o *avatar-IA*), y como consecuencia, cualquier actividad tiene más potencial para ser percibida.

El concepto «**AvatarRank**» se basa en una medida más intuitiva y directa: el «**AvatarRankScore**». Este concepto es similar al «**AvatarRank**» pero se aplica únicamente entre dos *avatares* cualesquiera y de manera inyectiva o unidireccional.

Pero antes de continuar en la definición vamos a caracterizar matemáticamente el concepto de *avatar* para trabajar con él. Partimos de la base que un *avatar* es un objeto perteneciente a un **V.I.E** que aparte de las características propias de todos los objetos cumple con una propiedad que llamaremos *rotation* con un valor numérico que representará un ángulo y que indicará en qué ángulo se encuentran los «ojos» de dicho *avatar*. Este ángulo será fundamental ya que la imagen que se renderiza en el cliente del usuario estará definido por «hacia dónde está mirando». Este ángulo *rotation* se expresa sobre el eje X-Y espacial, ignorando el eje Z para hacer la medida más antropomórfica. Expresado de manera formal:

$$A \in \Omega \in \mathbb{E}^3$$

$$A = \{\rho_1, \rho_2, \rho_3, \rho_R\} \quad / \quad \rho_i \in \mathbb{R}$$

y definimos la función  $rot(A)$  como la función rotación de un *avatar* de la siguiente manera:

$$rotation(A) = \rho_R$$

Y por lo tanto, siendo  $A_a, A_b$  dos avatares distintos entre si, definimos la función «**AvatarRankScore**» abreviada como  $f_{ARS}()$  como :

$$f_{ARS}(A_a \text{ over } A_b) = f_{ARS}(a \leftarrow b) = f_{ARS}(a, b) = S_{a,b}$$

Esta función se lee como «la puntuación de A sobre B», y se define como la cantidad de atención que el *avatar B* recibe del *avatar A*.

Una vez definido este término, volvemos al cálculo del «**AvatarRank**», que se deriva de una matriz de puntuaciones (scores) de todos los *avatares* a todos los *avatares*. Es decir, se calcula el «**AvatarRankScore**» entre todos los *avatares* presentes. Posteriormente se acumula el valor de las columnas de esa matriz.

A su vez, la puntuación base se calcula tomando dos medidas trigonométricas inmediatas: la distancia euclídea entre A y B (que notaremos mediante la función  $d_E()$ ) y el ángulo de la cara (que notaremos mediante la función  $rot()$ ) de A sobre B. En resumen, cuanto más B esté mirando a A, y más A y B estén cerca, más alto es la puntuación.



La función también necesita dos parámetros fijos que denominaremos  $D_{max}$  y  $D_{min}$  y que, respectivamente, indicarán la distancia (euclídea) máxima y mínima entre las cuales podrá operar el cálculo. Es decir, si dos avatares están a mayor distancia que  $D_{max}$  o a menor de  $D_{min}$ , no podrán ser evaluados.

Una vez establecidas estas premisas, se define la función «**AvatarRankScore**» abreviada tal que  $score(a, b)$  o  $S_{a,b}$  como:

$$f_{ARS}(a \leftarrow b) = S_{a,b} = \frac{D_{max} - (d_E(a, b) - D_{min})}{D_{max}} \cdot \frac{\Pi - |\arctan(b - a) - rot(b)|}{\Pi}$$

En la siguiente ecuación se define la función «**AvatarRank**», teniendo en cuenta que dicha función  $avatarRank(A)$  se aplica únicamente a un *avatar* e indica su «popularidad» tal y como se ha expresado anteriormente:

$$scoreMatrix_{N \times N} = \begin{bmatrix} S_{1,1} & S_{1,2} & \dots & S_{1,N} \\ S_{2,1} & S_{2,2} & \dots & S_{2,N} \\ \dots & \dots & \dots & \dots \\ S_{N,1} & S_{N,2} & \dots & S_{N,N} \end{bmatrix} \quad \forall i \in N, S_{i,i} = 0$$

$$avatarRank(X) = \frac{1}{N-1} \cdot \sum_{i=1}^N S_{X,i}$$

En las siguientes figuras se muestran algunas representaciones gráficas de la función  $score(a, b)$ :

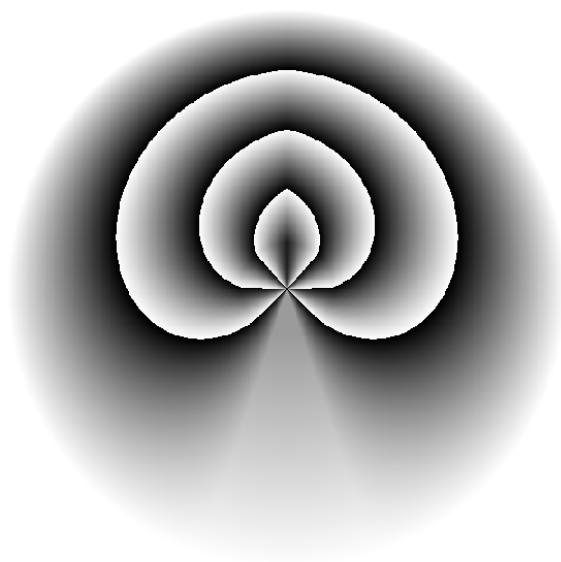


Figura 4.1: Función AvatarRankScore: representación espacial plana en un espacio cartesiano X-Y

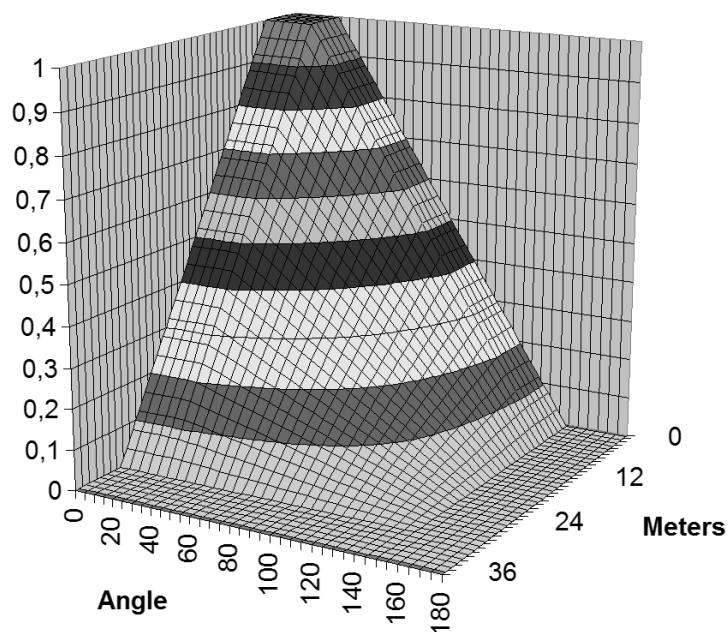


Figura 4.2: Función AvatarRankScore: representación lineal graduada

La función «**AvatarRankScore**» es una versión matemáticamente formulada del concepto antropomórfico de «espacio personal humano», como veremos más adelante.

En las líneas siguientes comentaremos algunos ejemplos de cómo interactúan los *avatares* con las funciones «*AvatarRankScore*» y «*AvatarRank*»:

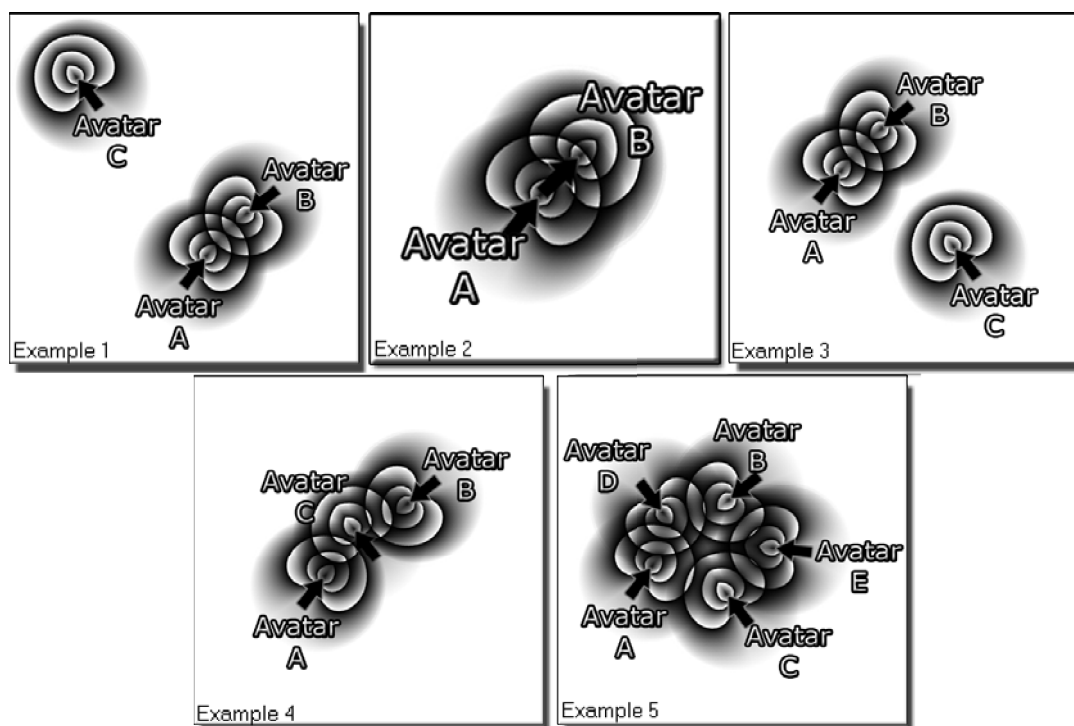


Figura 4.3: Ejemplos de AvatarRank

- **Ejemplo 1:** A y B están cerca, pero C está fuera de la cuestión.

*En este caso, A y B tienen un promedio de AvatarRank ya que están en la «zona sensible» de los demás. Por otro lado, C tiene un AvatarRank casi cero, ya que está en la zona no-body.*

- **Ejemplo 2:** El seguidor. A está siguiendo a B.

*A tiene un «AvatarRank» prácticamente a cero, pero B lo tiene muy alto. Cuando un avatar sigue a otro, el seguido suele estar en la zona «caliente» del seguido, pero el seguido no está en la zona «caliente» de nadie.*

- **Ejemplo 3:** A y B están cerca, pero C está algo interesado en A y B.

*En este caso, A y B también tienen un «AvatarRank» sobre la media, pero es un poco más alto que en el caso anterior, ya que C les está «mirando»*

- **Ejemplo 4:** El avatar «Alfa». A y B están fijándose en C, pero C no les está haciendo caso.

*Tanto A como B tienen un «AvatarRank» medio-bajo, pero C tiene un «AvatarRank» muy alto ya que está en la zona caliente de A y B*

- **Ejemplo 5:** La fiesta. Un grupo hablando en común. Todo el mundo ve a todo el mundo.

*Este es un comportamiento antropomórfico muy común. Algunos avatares se reúnen para tener una pequeña charla. Todos los avatares tienen un «AvatarRank» de nivel medio-alto, incluso cuando ningún avatar está completamente en una zona sensible. Cuando varios avatares están juntos y sus áreas sensibles se solapan, se crea un «hotspot». Cualquier avatar colocado en este hotspot verá disparado su «AvatarRank». Podríamos asimilar este comportamiento a un «corrillo» o a un «escenario» desde un punto de vista antropomórfico.*

En resumen, con estas funciones «AvatarRank» y «AvatarRankScore» estamos trabajando directamente sobre los siguientes objetivos de la presente tesis: G1, G2x, G3, G6, G9, G10, G11a

### 4.2.3 Desarrollos prácticos del experimento SL/2009

#### 4.2.3.1 Metabot Usher

Con el objetivo de disponer de un Metabot con un bajo nivel de realismo de manera voluntaria se le ha dado una apariencia metálica y unas capacidades interactivas muy básicas. Para darle un contexto se ha definido que este **Metabot** será el encargado de hacer las veces de «ujier» en las instalaciones virtuales de la Universidad. Se colocará automáticamente en el escritorio que está creado a tal efecto y esperará. Este escritorio se encuentra ubicado en la planta baja del edificio de la Universidad.

Cuando un *avatar* cualquiera entre en el recinto y se acerque a este **Metabot** deberá saludarle de una manera estándar y esperar a que hable.

Cuando un *avatar* se dirija al **Metabot** este hará una de estas tres cosas por orden:

1. Consultar en un fichero prefijado de preguntas y respuestas. Si la frase que se ha dicho está en ese fichero se le responderá con la respuesta prefijada en ese fichero de texto.
2. Si no está en ese fichero se consultará la agenda de la Universidad en la web y se buscará alguna coincidencia. Si se encuentra se responderá la información
3. En otro caso se indicará que no se comprende la respuesta.

El fichero de texto con la información predeterminada tendrá un aspecto similar a este ejemplo:

hola:hola muy buenos días, bienvenido a la Universidad Virtual  
adios:adios que tenga buen día, esperamos volver a verle pronto



Figura 4.4: Metabot Usher

Si bien es cierto que este **Metabot** es extremadamente sencillo, el simple hecho de detectar un concepto tan humano como «se acerca» o «se aleja», requiere de herramientas como el «**AvatarRank**» anteriormente descrita para poder implementar esta funcionalidad.

Conclusiones. Este **Metabot** es prácticamente el más simple que se puede construir, pero a pesar de esto, es fácil ver su potencial como canal de comunicación y filtrado que sirva de interfaz entre la información que podemos encontrar en la web y los **V.I.E's**. Otro objetivo importante es dar una interfaz de usuario antropomórfica al contenido web, siguiendo el **principio de máxima verosimilitud**.

El desarrollo de este **Metabot** incide en los siguientes objetivos de la presente tesis: *G1, G2x, G11a*

#### 4.2.3.2 Metabot Pretender

Este *Metabot* fue diseñado pensando en un nivel medio de verosimilitud. Aunque el atuendo externo del *avatar* es claramente antropomórfico se ha optado por mantener una piel irreal para un ser humano. Independiente del aspecto externo se buscaba darle cierta interacción inteligente, pero sin llegar a ser demasiado creíble.

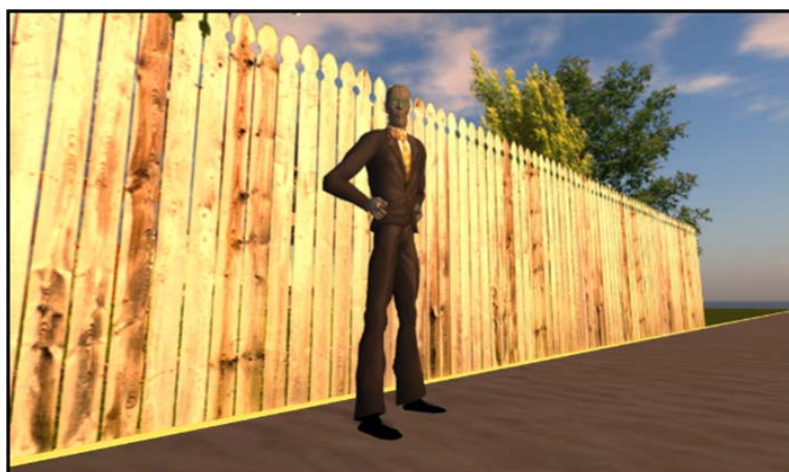


Figura 4.5: Metabot Pretender

Para ello se desarrolló el *Metabot* «pretender» capaz de experimentar sus habilidades de conversación utilizando motores de indexación. Si en el AIML las reglas lingüísticas se expresan cuidadosamente mediante enlaces sobre conjuntos de palabras, los experimentos con indexación apuntan a almacenar todas las frases que se van adquiriendo para luego acceder a estas frases en base a ciertas palabras que funcionan como *input*. Por ejemplo, si un *Metabot* escucha la frase «*el lunes es el primer día de la semana*», lo almacenará. Tiempo después escuchará «*semana*» y recuperará ese recuerdo, la frase completa, donde se dijo, quién la dijo, quién estaba cerca cuando ocurrió y cualquier otro dato con el que pueda realimentarse.

El algoritmo que sigue este *Metabot* es el siguiente:

- Cuando escucha algo en el **V.I.E** busca en su base de datos indexada
- Si encuentra algo con una puntuación alta lo «dice»
- En caso contrario consulta una frase célebre aleatoria de Internet, que al final también terminará indexada.

Este experimento pretende investigar un enfoque basado en la imitación pura, trabajando con grandes cantidades de datos y herramientas para acceder a esta información rápidamente. Es a la vez interesante e inquietante observar como cualquier sujeto del experimento que no esté al corriente de la naturaleza simplista de este *Metabot* puede estar conversando durante algunos minutos con frases vacías sin ningún tipo de comportamiento inteligente subyacente.

El desarrollo de este *Metabot* incide en los siguientes objetivos de la presente tesis: *G1*, *G2x*, *G9*, *G11a*

#### 4.2.3.3 Metabot Chatterbox

En el XVIII Premio Loebner de Inteligencia Artificial, realizado el fin de semana del 11 al 12 de octubre de 2008, Elbot convenció a 3 de los 12 interrogadores humanos de que era un ser humano, consiguiendo el Premio de Bronce. Además este motor de chat o «chatterbox» quedó publicado a través de su página en Internet <http://www.elbot.com>. Esta característica nos dio la idea de crear un cuerpo para Elbot dentro de *Second Life*: el *Metabot* chatterbox.



Figura 4.6: Metabot Chatterbox

Además se utilizó un *avatar* prefabricado con alto nivel de realismo para que, junto con unas habilidades conversacionales avanzadas, estuviésemos delante del *Metabot* más

realista hasta la fecha.

El algoritmo que se programó para este *Metabot* era similar al del *Metabot* «pretender», pero en lugar de responder a las frases escuchadas mediante un motor de búsqueda indexado, solicitaba la respuesta a Elbot a través de su servicio web.

El desarrollo de este *Metabot* incide en los siguientes objetivos de la presente tesis: *G1*, *G2x*, *G9*, *G11a*

#### 4.2.4 Conclusiones

Los *Metabots* mencionados anteriormente estuvieron funcionando 24 horas al día, 7 días a la semana durante 8 semanas completas dentro de un recinto público en *Second Life*, recopilando toda la información sobre las interacciones con los *avatares* manejados por usuarios de la plataforma.

La métrica más importante que se ha utilizado para estudiar los datos obtenidos es el número de «líneas de chat» intercambiados en una conversación entre un *Metabot* y un humano. Estos son los datos obtenidos en función del *Metabot* en cuestión.

Figura 4.7: Líneas de chat por metabot: usher, chatterbox o pretender

En este gráfico podemos ver que el *Metabot* «Usher» es con el que menos se ha hablado, el «Pretender» consigue tener más conversaciones de tres frases de longitud,



aunque sin duda es el «Chatterbox» el *Metabot* que consigue mantener conversaciones más largas, llegando a mantener la atención de los usuarios humanos durante más de 6 o 7 frases en numerosas ocasiones.

Si retomamos la hipótesis de partida de este experimento: *a mayor realismo gráfico e interactividad del avatar, mayor predisposición tendrán los usuarios a interactuar con dichos avatar-I.A's* parece que se cumple.

Basándonos en estos datos se realizó un segundo experimento como una simplificación del primero: manteniendo los mismos *avatares* se niveló su nivel de charla interactiva, utilizando en los tres casos el *Metabot* «usher». Los resultados se muestran en la siguiente tabla:

Figura 4.8: Líneas de chat por apariencia humana del avatar

Conclusión: La simple imagen del *avatar* puede afectar severamente a los resultados finales, revalidando la hipótesis de partida, y por lo tanto el principio de máxima verosimilitud y la metáfora del entorno.

Adicionalmente, y fuera del objetivo del experimento, al examinar la primera tanda de conversaciones nos percatamos de que algunas personas nunca se dieron cuenta de que no estaba hablando con una máquina: llegaron, hablaron con los *Metabots* y se fueron. Aunque este hecho no es constitutivo de un «Test de Turing», si resulta al menos interesante. Los resultados obtenidos indicaban que del 15 % de usuarios fueron «engañados» por el Chatterbox y 4 % por el «Pretender».

Figura 4.9: Porcentaje de usuarios engañados

Queda por tanto confirmada sobradamente la hipótesis de partida y de manera colateral los siguientes objetivos de la presente tesis: *G1, G2x, G6, G9, G10, G11a*

## 4.3 Experimento SL/2011

### 4.3.1 Introducción

Este experimento se llevó a cabo entre los años 2011 y 2012 aprovechando las materias y experimentaciones realizadas en el Máster de Ciencia y Tecnología Informática.

Este experimentos también fue llevado a cabo en la Isla TESIS dentro del **V.I.E Second Life**. El objetivo principal de dicho experimento era comprobar la **Hipótesis 9** *definición de servicios necesarios*.

La hipótesis refinada para este experimento fue expresada como sigue: *Partiendo de un V.I.E en el que se da un problema genérico de alto nivel deliberativo, es posible implementar la solución mediante un S.M.A.S utilizando una aproximación «servicio externo deliberativo» tal y como se describe en el punto 3.12.1.3.*

Para la ejecución de este experimento se requerían los siguientes elementos

- Un problema complejo genérico implementado en un entorno virtual:
  - Para ello se definió un «tablero» tipo ajedrez o damas donde los agentes pueden moverse.
  - Cada casilla del tablero puede: estar vacía, ocupada con un obstáculo inamovible, ocupada con un obstáculo que puede ser destruido.
  - Pero para destruir un obstáculo es necesaria la colaboración de dos o más agentes.
- Una serie de agentes capaces de realizar acciones simples: mover y golpear
  - Para ello se definió un prototipo de agente corpóreo en el **V.I.E** consistente en una esfera sencilla capaz de rodar.
  - A cada agente se le definió un código LSL para interpretar las órdenes del servicio externo.
- Un framework software que permitiese la implementación como un servicio externo deliberativo de dichos agentes.
  - Se utilizó una versión inicial del framework «**SSR-Hub**»

- Dentro del Hub se implementó el módulo particular para el implementar los servicios externos deliberativos necesarios para resolver problema indicado y se denominó «**Sprawl1**». En otras

Desde el punto de vista de la presente tesis, en este experimento se ha trabajado y/o demostrado directamente sobre los siguientes objetivos: *G1, G6, G7x, G8, G9, G10, G11b*

### 4.3.2 Estrategia Multiagente Deliberativa

En una primera fase de investigación se desarrolló el módulo de servicios externos Sprawl1 dentro del framework **SSR-Hub** como base para realizar experimentos con un agente BDI clásico dentro de un *Entorno Virtual Inmersivo*. Este framework se ha diseñado siguiendo el modelo teórico «despliegue-representación» aplicado a un problema concreto: la movilidad sincronizada de agentes. Se eligió este problema para ayudar a generar el framework y refinar el modelo teórico debido a su enfoque clásico: situation calculus, modelo BDI, fusión de datos, ontología expresada en PDDL y detección de símbolos directamente del entorno.

El problema tiene unas bases muy sencillas: existe un tablero sobre el cual se desenvuelven agentes que deben moverse de una casilla a otra según es indicado por el usuario. El tablero puede tener casillas libres, ocupadas por un agente, ocupadas por un obstáculo fijo u ocupadas por un obstáculo temporal. Cuando un agente determinado tiene la intención de ir a una casilla en concreto debe planificar sus movimientos e indicar a los demás agentes su colaboración si así es necesario.

En este caso existe una dicotomía clarísima derivada de la tecnología utilizada: toda la estructura de despliegue ha sido realizada en *Second Life* y LSL con una capacidad de cálculo reducida, mientras que la representación simbólica espacial con el planificador automático y la gestión de los agentes se ha realizado en Java. Dentro de esta capa, toda la representación cognitiva de los agentes se realiza mediante PDDL y la comunicación mediante KQML.

El resultado ha sido un sistema extremadamente robusto y funcional incluso con los habituales problemas provenientes de la discretización del espacio continuo del simulador.

A continuación se describen en profundidad los detalles técnicos de la herramienta.

### 4.3.3 Arquitectura Sprawl1

En este sistema se han utilizado técnicas propias de sistemas MultiAgente que lo hacen muy interesante desde el punto de vista de la experimentación. Algunas de estas características son:

- Modelo BDI (Beliefs-Desires-Intentions)
- Fusión de datos obtenidos por los diferentes agentes (fusión de Beliefs)
- Ontología automáticamente generada basada en dominio PDDL
- Sensorio simbólico

El resultado ha sido un sistema altamente flexible y robusto, aunque con mucho camino por recorrer. Es posible definir y modificar escenarios y problemas de manera interactiva tridimensional. La suma de todos estos factores hace que las posibilidades en investigación y docencia sean enormes.

#### 4.3.3.1 Descripción del problema: Objetivo

El objetivo principal es el siguiente: dado un escenario inicial se requiere que un agente móvil sea capaz de desplazarse desde un punto de origen hasta un punto destino.

Hay que tener en cuenta que al estar trabajando dentro de *Second Life* nos estamos refiriendo en todo momento a una configuración de objetos físicos simulados. Sin embargo para poder describir el problema formalmente hay que definir todos los posibles elementos del mismo:

- **TILES:** O también podríamos llamarlos baldosas. Estos objetos con forma de baldosa de 1x1x0.1 metros se colocan en forma de tablero de ajedrez. Para que un objeto sea considerado como TILE por los agentes su nombre debe comenzar por estas cinco letras «TILE\_». Por ejemplo TILE\_1 TILE\_A TILE\_ZZZZ son ejemplos válidos. Estos objetos son muy importantes ya que los agentes los usan como símbolos para relacionar posiciones espaciales con elementos de la ontología. De hecho el elemento más abundante en las ontologías son los TILES. En la implementación, para facilitar su comprensión, los TILES tienen nombres del tipo

TILE\_X\_Y donde  $X,Y=\{0,1,2,3,4,\dots\}$ . Siendo X e Y su posición en el tablero. En realidad a los agentes no les importa esas coordenadas ya que no hacen ningún tipo de cálculo en base a su posición, sin embargo es más sencillo para los humanos ver una matriz como pares de elementos X,Y que un array de  $X*Y$  elementos.

- **WALLs:** O muros. Son bloques de  $1x1x1.5$  metros que se sitúan exactamente sobre un TILE. Si hay un WALL sobre un TILE el agente no podrá pasar por ese TILE. Los WALLs no pueden ser eliminados ni movidos.
- **OBSTACLES:** Son bloques de  $1x1x1.5$  que a diferencia de los WALLs pueden ser retirados. Los OBSTACLES tienen un factor de resistencia  $\leq 10$ . Si la resistencia de un OBSTACLE llega a 0, el OBSTACLE desaparece por completo. Cada vez que un OBSTACLE recibe un golpe se resta un punto a la resistencia. Cada segundo esa resistencia se incrementa en +1. Esto significa que necesita más de 1 golpe por segundo para terminar siendo eliminado. Esto se puede conseguir únicamente si dos o más agentes golpean el OBSTACLE a la vez.
- **AGENTES:** Los agentes son también parte del escenario del problema. Tienen la apariencia física de una esfera pero lo importante es el script que tienen dentro. Este script es capaz de ofrecer movilidad al objeto. Por otro lado, un AGENTE no puede ocupar el TILE ocupado por otro AGENTE al igual que un WALL o un OBSTACLE. Sin embargo si puede solicitar al AGENTE que moleste que se mueva, o mejor dicho, que cambie su posición a otro TILE.

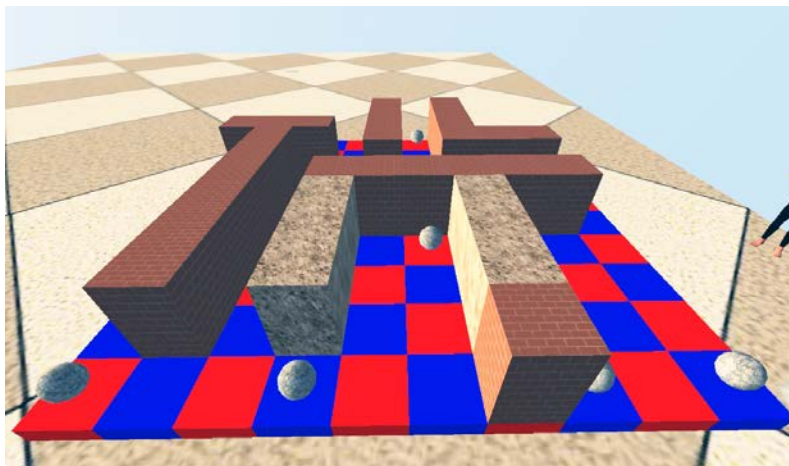


Figura 4.10: Ejemplo de una posible configuración física del problema

Cada escenario se genera «a mano». El tablero se genera automáticamente con dos comandos de voz que se le da al agente generador del tablero:

- «drop» elimina todo el tablero
- «build X Y» genera un tablero de X Y

Después la colocación de **WALLS**, **OBSTACLES** y **AGENTES** se hace a mano.

Hay que tener en cuenta que incluso cuando partimos de un mismo escenario podemos tener muchos problemas distintos, ya que podemos elegir cualquier agente y cualquier TILE destino.

Para lanzar el problema solamente hay que indicar a un AGENTE que incluya en su lista de deseos estar en el TILE que indiquemos. Esto se realizará con los siguientes pasos:

- Tocamos un **AGENTE** cualquiera. En ese momento esperará escuchar por el canal de chat el nombre de un **TILE**. Si en los siguientes 30 segundos escucha una cadena de caracteres que comienza por «**TILE\_**», tomará esa cadena como objetivo.
- Podemos decir por el canal de chat el **TILE** al que queramos que el AGENTE se mueva o aún más fácil, tocar el **TILE** que deseemos. Los **TILES** tienen un pequeño script que hace que cada vez que se toca digan su nombre por el canal de chat.

#### 4.3.3.2 Descripción del problema: Actores participantes

Existen tres actores en este sistema: Humano(s), Agente0 y AgentesX

**Actor humano** Dado que estamos inmersos en un mundo tridimensional virtual fuertemente antropomórfico (metaverso) como es *Second Life* es lógico que el origen de las intervenciones que desencadenan comportamientos sea humano. De hecho para poder experimentar (ver) lo que está ocurriendo en el sistema hemos de acceder a nuestro simulador *Second Life* mediante un avatar. Por este simple hecho podremos interactuar con el sistema con el interfaz humano: tocando, hablando y escuchando.

Como detalle adicional, todos los mensajes FIPA-ACL (en realidad un subconjunto de KQML) que intercambian los agentes se transmiten a través del canal de chat,

por lo que podemos «escuchar» los mensajes directamente. No solo eso, además si «hablamos» directamente en el lenguaje adecuado podemos hacernos pasar por un agente e interactuar a nivel de intercambio de mensajes. Por ejemplo, si decimos «(stream-about :name ag0:receiver ag1 :reply-with 123 :ontology sprawl1 :lenguaje pddl)» el agente 1 responderá de igual manera que cuando se lo solicita el agente 0.

Pero lo más importante que debe hacer un humano para experimentar en el sistema es tocar. Como se puede observar en el autómata de estados más adelante en este documento, un AgenteX estará en reposo hasta que pasen dos cosas: que otro agente le pide una acción o que un humano le toca.

**El Agente 0** Otro actor fundamental en el sistema Sprawl1 es el Agente 0. Este agente es muy especial por varias razones:

- Sólo existe una instancia de Agente 0 en un sistema Sprawl1.
- El Agente 0 está programado en Java y se conecta a **Second Life** a través de un Avatar como lo hace un humano gracias al framework **SSR-Hub**.
- Es la base del sistema Sprawl1, ofreciendo varios servicios de plataforma: AMS (Agent Management Service), Fusión de Datos (beliefs), Planificación, Interfaz Web, etc. (se describen por completo en el siguiente punto)
- El Agente 0 NO sigue el modelo BDI, ya que en realidad es un interfaz de comunicaciones con la capa de nivel alto de abstracción o «capa de representación simbólica espacial».
- En cuanto a su interacción dentro de **Second Life**, el Agente 0 sólo interactúa por el canal de chat, no toma ningún dato de su sensorio ni realiza movimientos.

Por lo tanto podemos decir que si no existe Agente 0 no existe Sprawl1, aunque tengamos muchos Agentes X. Esto es un problema para la robustez del sistema, ya que si cae el Agente0 cae todo el sistema. Una posible evolución del sistema sería poder traspasar el grueso de los servicios a los Agentes X y que sean capaces de acceder por sí mismos a la capa de representación simbólica espacial.

La razón fundamental para agrupar tantos servicios en el mismo Agente tiene su origen en la limitación de la plataforma. Podríamos decir de manera general que los



Agentes diseñados en LSL (los Agentes X) son muy buenos a la hora de desenvolverse en el mundo virtual ya que están hechos de su misma materia, pero tienen limitaciones severas en cuanto a memoria y capacidad de cálculo. Por el contrario, el Agente 0 se desenvuelve más lentamente en el entorno virtual, pero como ventaja, al estar diseñado en un lenguaje muy potente (Java) y estar corriendo su ejecución en un equipo externo al simulador posee una capacidad de cálculo virtualmente ilimitada.

Como uno de los objetivos principales de este trabajo es dotar a los agentes de un plan base como esqueleto de las Intentions generado con un planificador automático clásico, y estos consumen gran cantidad de CPU y memoria opté por construir un agente (el 0) con nada de actividad «física-motriz» y toda actividad «mental-cómputo» en contrapartida de los múltiples Agentes X que están mayormente orientados al movimiento y tienen poca capacidad de cálculo.

Una característica interesante desde el punto de vista de la arquitectura es la mensajería multithread que posee el Agente0. Dado que el Agente0 debe procesar mensajes en paralelo de varios agentes al mismo tiempo está diseñado para procesarlos en paralelo y no bloquear así la gestión de múltiples agentes.

**Los Agentes X** Por último, tenemos a los Agentes X. Estos agentes están modelados siguiendo la arquitectura BDI. Existe una correspondencia directa entre estos tres conceptos y su representación en la programación del Agente X:

- **Beliefs:** Sensorio simbólico + memoria tiles (lista tilemem)
- **Desires:** Una vez un humano indica el TILE al cual debe ir se marca el Goal, o la expresión (bot-on-tile AGX TILE\_X\_Y) que será el Goal del problema en PDDL
- **Intentions:** Autómata de estados + plan (conjunto de acciones planificadas)

Todos los Agentes X son instancias del mismo software. En *Second Life* el software se representa mediante scripts realizados en LSL (Linden Script Language). Dicho esto, la representación física que he dado a los Agentes es una esfera de color grisáceo. He elegido una esfera porque es el elemento geométrico más sencillo en cuanto a movilidad, ya que rueda en cualquier dirección. Sin embargo lo que realmente conforma un AgenteX es el script que lleva dentro, por lo que sería trivial hacer que cualquier objeto más complejo y con distinto aspecto formen parte del sistema. Esto podría llevar a engaño, ya que

nuestra visión antropomórfica nos dice que dos objetos son distintos si tienen distinta forma, y si viésemos un agente esfera, otro agente con forma de cubo y otro con forma de robot con antenas seguramente pensaríamos que son distintos agentes y que el robot con antenas es el más capaz. En realidad si los tres objetos tienen el mismo script serían completamente similares en capacidad.

Cuando un AgenteX es instanciado en el mundo virtual lo primero que intenta es registrarse con un Agente0. Es decir, los agentes X necesitan un servicio de registro para obtener un identificador válido. En el punto «protocolo de registro» se ve un ejemplo de este mecanismo.

Una vez registrado, el agente pasa a regir su comportamiento por el autómata de estados. Este autómata de estados, junto con el conjunto de acciones que conforma el plan modelan perfectamente las Intentions de los agentes.

Me gustaría remarcar que los agentes tienen dos maneras de iniciar su actividad motriz: por indicación de un actor Humano o por indicación de un actor AgenteX. En ningún caso el Agente0 va a indicar a un AgenteX que se mueva ya que el Agente0 no sigue el modelo BDI.

Volviendo a las dos maneras de iniciar la actividad, también son muy diferentes: si es un actor Humano el que toca al agente, éste tomará la indicación de objetivo como su Desire, por ejemplo, si toco al Ag3 y después al TILE\_1\_3, el Agente 3 definirá como Desire esta expresión (*bot-on-tile AG3 TILE\_1\_3*).

Por el contrario, si la actividad motriz es iniciada como petición concreta de otro agenteX no lo tomará como un Desire, sino como una Intention temporal o sub-goal. Es decir, si tenemos dos agentes AG1 y AG2, y el Desire de AG1 (*bot-on-tile AG1 TILE\_0\_0*) y en el plan el AG1 se encuentra con esta orden (*move AG2 TILE\_0\_0 TILE\_1\_0*) –por lógica, si AG1 quiere estar donde está AG2, AG2 tendrá que quitarse- llegará un momento en que AG1 pedirá a AG2 que se mueva de 0\_0 a 1\_0. En este caso, el Desire es de AG1, no de AG2. AG2 puede acceder o rechazar la petición de AG1 en función de sus propios Desires, sin embargo que acceda a la petición de AG1 no significa que tenga que adoptarlos. AG2 cumplirá (o no) el movimiento que le solicita AG1 pero nada más.

Obviamente, en estos casos pueden aparecer conflictos ¿qué ocurre si AG1 pide a AG2 que haga algo y AG2 dice que no? Precisamente aquí están las Intentions: tomando como base el autómata de estados y el plan actual tomará la decisión de actualizar sus Beliefs y replanificar o abandonar su Desire.

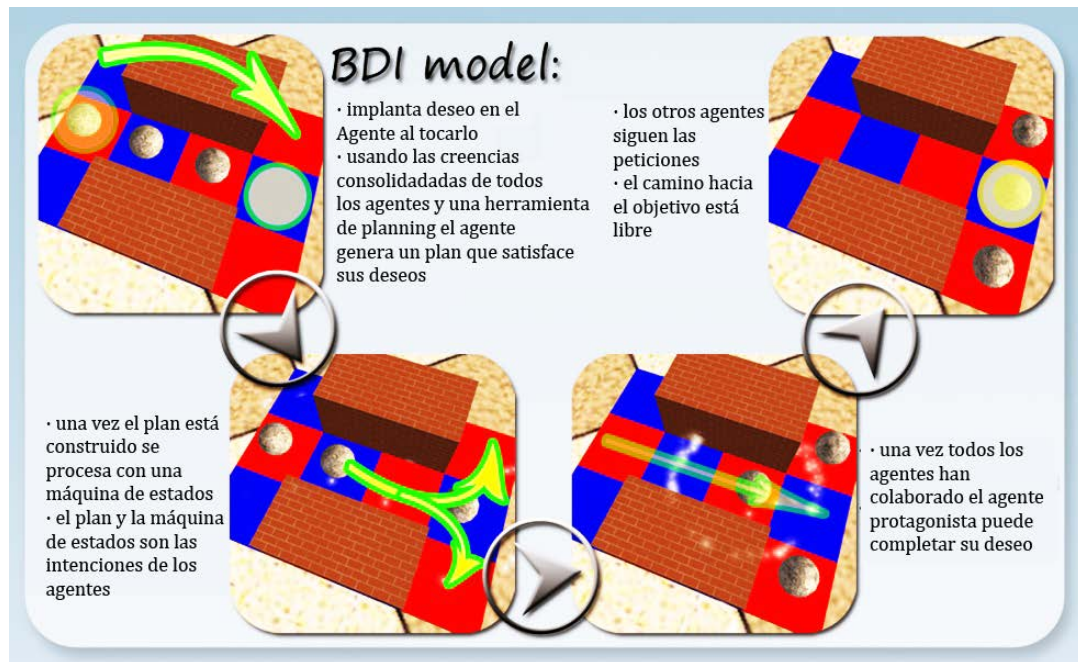


Figura 4.11: Modelo BDI

Por lo tanto, un Agente X al que un humano marca un objetivo (Desire) perseguirá hacer verdadera esa expresión de deseo hasta que lo consiga o decida abandonar por considerarlo imposible.

Hasta ahora he mostrado las capacidades más deliberativas de los Agentes, sin embargo no hay que olvidar que también posee ciertas capacidades reactivas aunque estas están imbuidas dentro de las Intentions:

- Si pasan más de 30 segundos entre una petición a otro Agente y su notify ya sea aceptando o denegando supondrá que el agente no está ya ahí y avanzará al siguiente paso de la planificación
- Romper un bloque OBSTACLE es una tarea especialmente física ya que consiste en colisionar repetidas veces contra el mismo obstáculo y a priori es imposible saber el tiempo que se va a tardar. Para detectar si se ha roto el OBSTACLE existen dos estados especiales en el autómata de estados que, utilizando feedback ante la aplicación de impulsos motrices, detecta si la posición actual del agente indica que el bloque se ha roto ya o no. Lo que en lógica de predicados es una acción (break AG1 TILE..) se convierte en este caso en un bucle controlado por métricas de distancias

- Moverse también es una tarea física y es posible que un Agente se quede bloqueado por algo que no había previsto (otro agente moviéndose, un avatar que pisa encima de los tiles, un fallo en los sensores, etc. . . ). Para ello he desarrollado también mediante feedback motiviz el control de «Stuck» (atascado en inglés). Este control también está imbuido en el autómata de estados, y puede desencadenar replanificaciones e incluso el abandono del Desire en determinadas condiciones.

Una característica muy interesante de los AgentesX es que, como buenos agentes mayormente deliberativos, son capaces de trabajar mediante lógica simbólica a partir de su sensorio. He denominado a esta herramienta «sensorio simbólico». Mediante el uso de esta técnica es posible construir la ontología del problema dinámicamente, de modo que se adapte a la realidad (Beliefs) de los Agentes.

Los AgentesX tienen una memoria reducida y actualizan su sensorio cada 15 segundos. En cada «revisión» del sensorio realiza los siguientes pasos:

- Para cada objeto que detecta y cuyo nombre comienza por «»
  - Almacena (o actualiza) sus coordenadas físicas en su memoria reducida [`«TILE_0_0»`, `<34,22,3334>`, `«TILE_1_0»`, `<35,22,3334>`, . . . .]
  - Analiza las coordenadas y distancia relativa entre TILES para establecer relaciones de adyacencia (*tile-is-adjacent* `TILE_0_0` `TILE_1_0`) y así generar la ontología dinámica que formará parte del problema PDDL a planificar llegado el momento.
- Para cada objeto que detecta y cuyo nombre comienza por «SprawlObstacle»
  - Analiza las coordenadas del obstáculo y determina el TILE sobre el que se encuentra (el TILE más cercano). Con esta información genera los predicados (*tile-with-obstacle* `TILE_X_Y`)
- Para cada objeto que detecta y cuyo nombre comienza por «SprawlWall»
  - Analiza las coordenadas del wall y determina el TILE sobre el que se encuentra (el TILE más cercano). Con esta información genera los predicados (*tile-with-wall* `TILE_X_Y`)

- Con la información que tiene sobre su posición actual y las coordenadas de los TILES busca sobre qué tile está situado y genera el predicado (*bot-on-tile AGX TILE\_X\_Y*)

Todos estos predicados conforman el sistema de creencias (Beliefs) que posee el AgenteX. Estos Beliefs quedan siempre almacenados en la pequeña memoria del AgenteX a la espera que Agent0 se los solicite. Es importante resaltar que estos Beliefs no representan casi nunca la totalidad del escenario del problema, ya que el sensorio de los objetos es limitado. Este hecho hace que intentar realizar un plan de tamaño mediano únicamente con el sensorio de un agente es normalmente inútil, para ello el Agente0 posee la capacidad de unificar los Beliefs de todos los agentes y conformar una consolidación sobre la que se genera el problema a planificar. Esta unificación de todos los Beliefs se ha denominado Aleph.

Como curiosidad, aunque estamos utilizando un simulador determinista, el motor físico del simulador que rige las actividades motrices dentro del mundo virtual rompe este determinismo incluyendo un grado variable de incertidumbre en las acciones físicas. Es decir, cuando un agente quiere moverse a unas coordenadas no tiene seguridad del tiempo que tardará en llegar, ni si llegará al lugar donde quiere, ni siquiera que consiga moverse, lo cual obliga a programar el sistema de Intentions con una robustez reactiva que se acerca a la realidad.

#### 4.3.3.3 Interfaz Web

Aunque todo el sistema está orientado a trabajar a través de *Second Life*, en ocasiones es más sencillo recibir información directamente del Agente0 a través de un medio clásico como es un navegador. Para ello, el Agente0 implementa un servidor web adhoc que es capaz de entregarnos una página web dinámica con la representación bidimensional del Aleph del agente0 en tiempo real. Además ofrece cierta interfaz muy básico para reiniciar el sistema. Este es un ejemplo de la información que ofrece por este canal:

#### 4.3.3.4 Recursos y su relación con servicios y actores

Los recursos del sistema son únicamente:

- TILES

























	0	1	2	3	4	5	6	7	8	9
0			AG9				AG4			AG2
1				Obs			Obs			
2				Obs			Obs			
3				Obs	AG11		Obs	AG6		
4										AG20
5							AG1			
6	AG10				AG14					
7										
8										
9	AG7				AG12					AG5

Figura 4.12: Ejemplo de interfaz de usuario

- WALLs
- OBSTACLEs

Estas relaciones vienen marcadas por el dominio del problema (ver punto Ontología) pero se pueden resumir así:

- Los AgentesX pueden moverse a un TILE si el TILE no tiene encima nada (ni un Agente ni un OBSTACLE ni un WALL)
- Los OBSTACLEs pueden romperse con la colaboración de dos robots
- Las WALLS no pueden romperse
- Un TILE puede tener sólo una de estas condiciones en un momento dado:
  - No tener nada encima
  - Tener un Agente
  - Tener un WALL
  - Tener un OBSTACLE



importante es la definición del dominio, es decir la definición de predicados. En este caso he utilizado un dominio que he llamado «BotGrid» y que reproduzco a continuación:

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; BotGrid
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;bots can move along !with-bot & !with-obstacle & !with-wall
;obstacles can be broken through
;walls can not be broken
;with-bot with-obstacle and with-wall are excludent between them

(define (domain BOTGRID)
  (:requirements :strips)
  (:predicates
    (is-the-same-bot ?x ?y)
    (bot-on-tile ?x ?y)
    (tile-with-bot ?x)
    (tile-with-obstacle ?x)
    (tile-with-wall ?x)
    (tile-is-adjacent ?x ?y)
  )

;x bot
;y tile from
;z tile to
  (:action move
    :parameters (?x ?y ?z)
    :precondition ( and
      (not (tile-with-obstacle ?z))
      (not (tile-with-wall ?z))
      (tile-is-adjacent ?y ?z)
      (bot-on-tile ?x ?y)
      (not (tile-with-bot ?z))
    )
    :effect ( and

```



```

        (not (bot-on-tile ?x ?y))
        (tile-with-bot ?z)
        (bot-on-tile ?x ?z)
        (not (tile-with-bot ?y))
    )
)

;v bot1
;w bot2
;x tile-bot1
;y tile-bot2
;z tile to break
(:action break
  :parameters (?v ?w ?x ?y ?z)
  :precondition ( and
    (not (is-the-same-bot ?v ?w))
    (bot-on-tile ?v ?x)
    (bot-on-tile ?w ?y)
    (tile-with-obstacle ?z)
    (tile-is-adjacent ?x ?z)
    (tile-is-adjacent ?y ?z)
  )
  :effect ( and
    (not (tile-with-obstacle ?z))
  )
))

```

Este dominio ha sido diseñado y probado para ser aceptado por el planificador clásico automático Metric-FF <http://www.loria.fr/~hoffmanj/2002.html>. Dicho planificador ha sido integrado dentro del sistema como un servicio.

La idea que he perseguido a lo largo de todo el trabajo es hacer coincidir toda la lógica simbólica que subyace a un agente deliberativo (y en especial a los que siguen el modelo DBI) con este dominio. En otras palabras, utilizar una ontología expresada en

PDDL y siguiendo las reglas de un dominio que permitiese su conversión directa a un problema entendible por Metric-FF.

En cualquier caso, dejando a un lado el objetivo de poder dar un servicio de planificación clásica, los predicados y acciones que manejan los Agentes son:

- Predicados:
  - (bot-on-tile ?agent ?tile)
  - (tile-with-obstacle ?tile)
  - (tile-with-wall ?tile)
  - (tile-is-adjacent ?tile1 ?tile2)
- Acciones:
  - (move ?agent ?tile-from ?tile-to)
  - (break ?agent1 ?agent1 ?tile-to-break)

#### 4.3.3.7 Ejemplo de diálogos FIPA-ACL (KQML)

Para dar una visión más contextual se van a reproducir aquí algunos de los diálogos en FIPA-ACL (KQML) que se intercambian en el sistema:

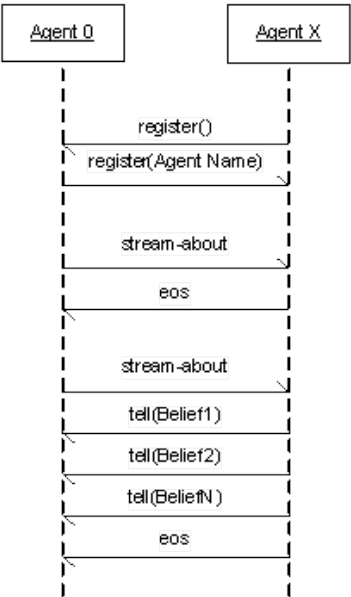


Figura 4.14: AgenteX se registra en el Agente0

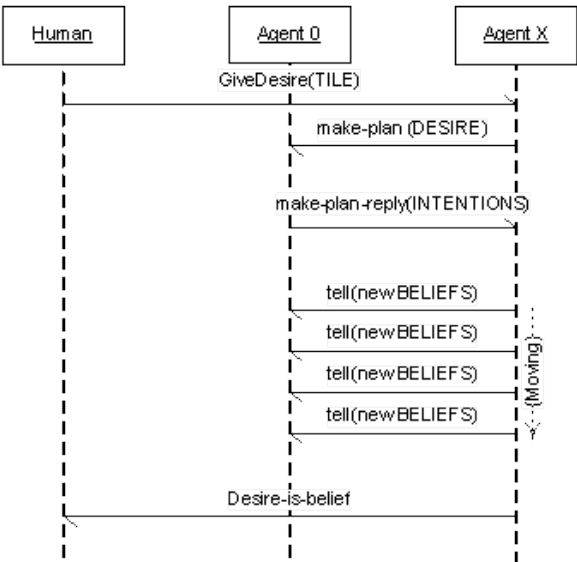


Figura 4.15: AgenteX se mueve sin necesidad de colaboración

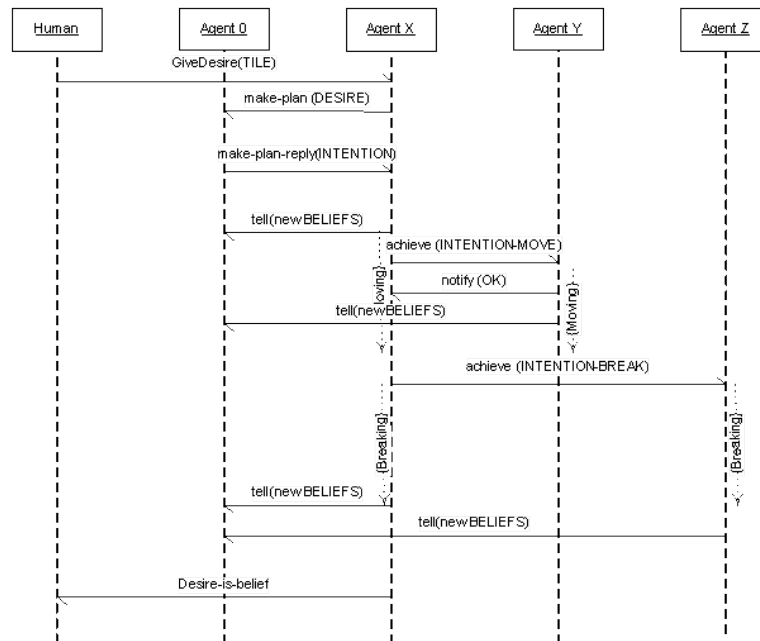


Figura 4.16: AgenteX mueve y rompe obstáculo con ayuda de Y y Z

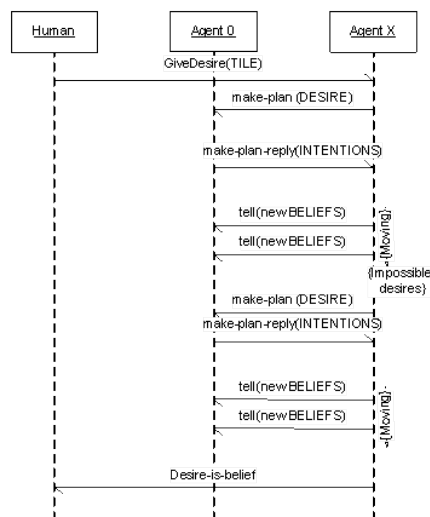


Figura 4.17: AgenteX se mueve pero tiene que replanificar

### 4.3.4 Experimentación

Una vez sentadas las bases de la arquitectura se ejecutó el experimento cuatro veces, cada una con una configuración de problema distinta, con el objetivo de poner a prueba la hipótesis inicial de trabajo. Hay que matizar que cuando indicamos que se realizaron «cuatro configuraciones distintas» únicamente se cambiaron las piezas en el tablero. Siempre se utilizaron las mismas reglas PDDL, el mismo FIPA-ACL y el mismo código tanto en los Agentes-X como en el Agente-0 y en el **SSR-Hub**.

En este vídeo se explica cómo funciona el entorno del problema: <https://www.youtube.com/watch?v=Wp0515jhmrU>

#### 4.3.4.1 Problema 1. Pasar por la puerta

En esta configuración del tablero cuatro agentes están distribuidos de manera no bloqueante, pero existe una pared con un solo espacio para pasar y los demás agentes deberán retirarse para dejar pasar al primero.

El agente debe moverse desde la posición con la flecha verde a la posición con la flecha amarilla.

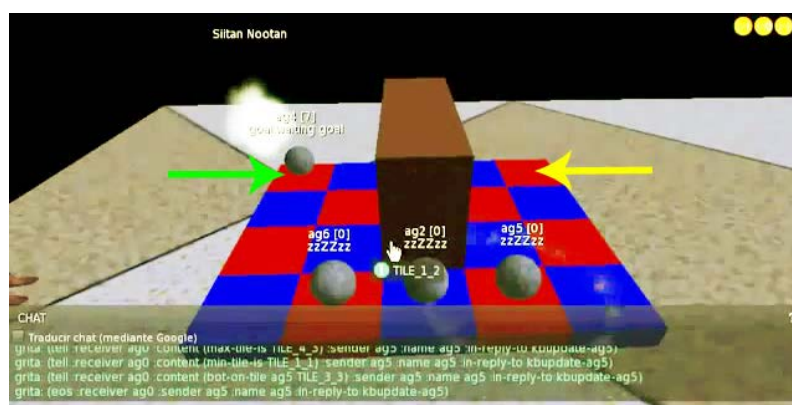


Figura 4.18: Pasar por la puerta

Se puede observar un vídeo de dicho experimento en <https://www.youtube.com/watch?v=16G2ULnXoqM>.

**Resultado:** No existe ningún contratiempo. Los agentes se retiran siguiendo el plan diseñado con el «problem solver». Podemos confirmar que el experimento es exitoso.

#### 4.3.4.2 Problema 2. El pasillo

En esta configuración del tablero cuatro agentes están distribuidos de manera bloqueante a lo largo de un pasillo. Según el modelo PDDL, dos agentes no pueden estar en el mismo TILE al mismo tiempo, pero en la realidad sí es posible. El problema se divide en dos casos, una tarea soluble y otra insoluble.

Tarea 1: El agente debe moverse desde la posición con la flecha verde a la posición con la flecha amarilla.

Tarea 2: El agente debe moverse desde la posición con la flecha verde a la posición con la flecha rosa.

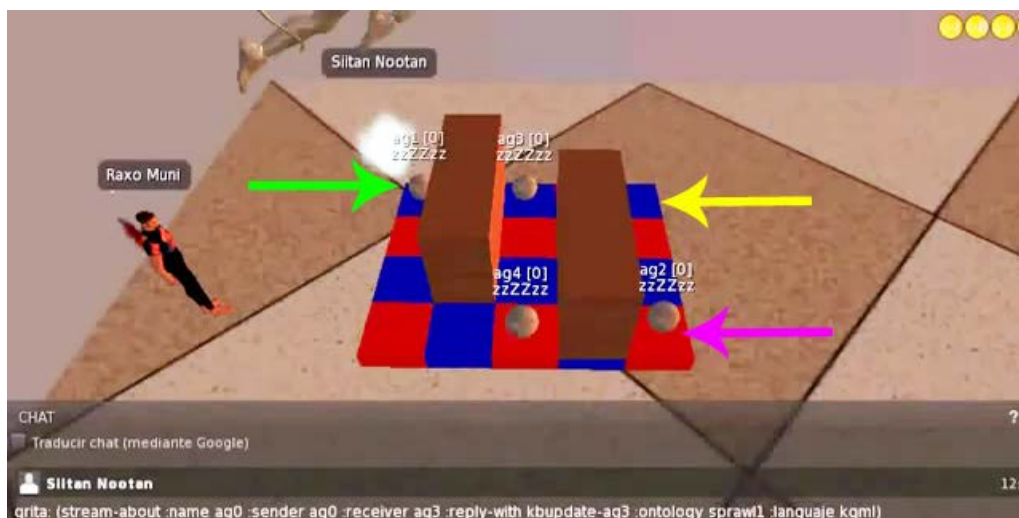








Figura 4.19: El pasillo

Se puede observar un vídeo de dicho experimento en <https://www.youtube.com/watch?v=1twDk640H1s>

**Resultado:** En la tarea 1 no existe ningún contratiempo, pero obviamente la 2 resulta complicada ya que el planificador no lo permite. Sin embargo hemos de recordar que estamos implementando un **S.M.A.S**, lo que significa que los agentes replanifican sobre la marcha y son adaptativos, por lo que en algunas ocasiones, y dependiendo de la sincronización de los agentes, la tarea 2 también se cumplía. Podemos confirmar que el experimento es también exitoso.

	0	1	2	3	4
0	AG1		AG3		
1					
2					
3			AG4		AG2

Agentes: XXXX  
 : 123456789012345678901234567890123  
 : 11111111112222222222223333

Figura 4.20: El pasillo según la percepción conjunta de los agentes

#### 4.3.4.3 Problema 3. El conflicto

En esta configuración del tablero cuatro agentes están distribuidos de manera bloqueante a lo largo de un pasillo compartido. En un momento dado se indica a dos agentes distintos (flecha verde) que se sitúen en el mismo TILE (flecha amarilla), lo cual parece que desembocará en un conflicto.

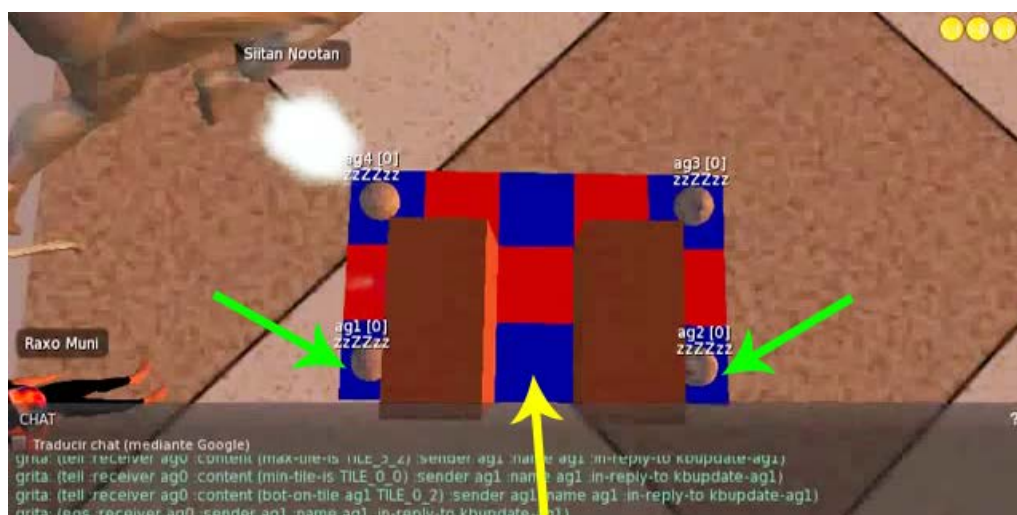


Figura 4.21: El conflicto, paso 1

Se puede observar un vídeo de dicho experimento en <https://www.youtube.com/>



Figura 4.22: El conflicto, paso 2



Figura 4.23: El conflicto, paso 3

watch?v=Yp1S6Rm4lww

**Resultado:** Aunque las reglas PDDL dicen que los TILES no pueden albergar dos agentes al mismo tiempo, el sistema de creencias individual de cada agente permite casos como los que se ven en el paso 2. En realidad cada agente pide una planificación por separado, cada uno con los datos que tienen en el momento inicial. Siguiendo cada uno con su plan, y en base a la pequeña capacidad reactiva de estos agentes, son capaces de resolver el problema. Si bien es cierto que no siempre ocurre en la misma secuencia ya que existen condiciones de carrera en la planificación que añaden imprevisibilidad al



resultado y a veces todos los agentes se quedan atascados. Por esta razón podemos decir que el sistema se comporta medianamente bien cuando las planificaciones se hacen secuencialmente, pero se vuelve imprevisible cuando se realizan planificaciones simultáneas.

#### 4.3.4.4 Problema 4. La cooperación

En este último problema la configuración es algo distinta. Sólo existen tres agentes, y en lugar de muro inamovible, existe un obstáculo que divide el tablero en dos. El objetivo de un agente es pasar a la zona bloqueada del tablero, y para ello requiere de la colaboración de otros agentes.

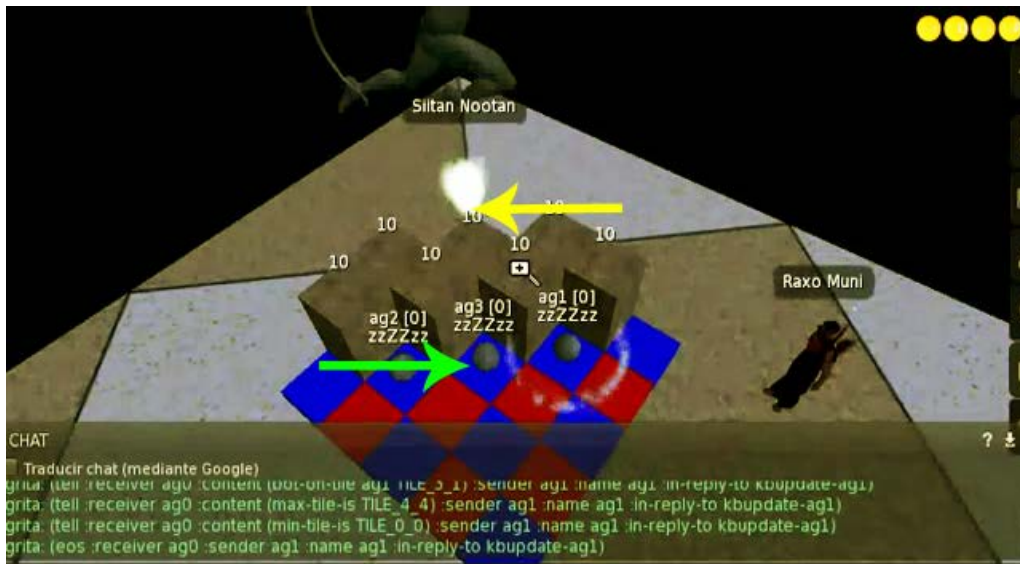


Figura 4.24: La cooperación

Se puede observar un vídeo de dicho experimento en <https://www.youtube.com/watch?v=McrPoNki2Qs>

**Resultado:** En este caso la resolución del problema es un éxito completo.

	0	1	2	3	4
0			Obs	Obs	
1		Obs	Obs	AG1	
2	Obs	Obs	AG3		
3	Obs	AG2			
4					

Agentes: XXX  
: 123456789012345678901234567890123

Figura 4.25: La representación de la percepción de los agentes



Figura 4.26: Detalle del momento en el que los agentes colaboran

### 4.3.5 Conclusiones

Dado que el experimento no tiene unas características cuantitativas concretas no es posible mostrar unas métricas. Pero si recordamos la hipótesis de trabajo:

*Partiendo de un **V.I.E** en el que se da un problema genérico de alto nivel deliberativo, es posible implementar la solución mediante un **S.M.A.S** utilizando una aproximación «servicio externo deliberativo» tal y como se describe en el punto 3.12.1.3.*

El objetivo es comprobar si es realmente viable la implementación de servicios deliberativos de alto nivel de complejidad en **V.I.E's** mediante **S.M.A.S's**, lo que queda de sobra probado. Por tanto la hipótesis queda comprobada e indirectamente también los siguientes objetivos: *G1, G6, G7x, G8, G9, G10, G11b*

## 4.4 Experimento OS/2012

### 4.4.1 Introducción

Este experimento se llevó a cabo entre los años 2012 y 2013 y es una de las experimentaciones realizadas mediante el curso de Doctorado más interesantes.

Al contrario que los experimentos anteriores se cambió la plataforma de provisión de entorno de *Second Life* a *Open Simulator* por las siguientes razones:

- En este experimento la participación del público general no era necesaria
- El rendimiento de los agentes en la «estructura de despliegue» era fundamental y al poder ejecutar *Open Simulator* en una plataforma propia y dedicada era posible darle mucha más capacidad de cómputo.
- El tiempo de latencia de red también era un problema. Los servidores de *Second Life* podrían estar en cualquier lugar del mundo, y la distancia entre estos servidores y el **SSR-Hub** ( $ch\beta$ ) no debería ser excesivamente grande (ver punto 3.5.6).

Por lo tanto en esta ocasión los experimentos fueron llevados a cabo en un **V.I.E** con las mismas características técnicas que los anteriormente ejecutados en *Second Life* pero dentro de un laboratorio cerrado al acceso libre y con una metodología más controlada.

Este experimento continúa en parte lo investigado en 4.3 en la parte que encontramos una limitación a la hora de realizar planificaciones automáticas concurrentes. Los agentes que se implementaron en ese experimento eran de tipo deliberativo y no eran capaces de manejar correctamente algunas de las situaciones inesperadas que acaecían. En esta ocasión el objetivo principal también es comprobar la **Hipótesis 9** *definición de servicios necesarios* caracterizada como sigue:

*Una vez se ha validado la capacidad de la arquitectura propuesta para implementar S.M.A.S deliberativos, es el momento de, partiendo también de un V.I.E en el que se da un problema genérico de alta complejidad tanto deliberativo como reactivo, es posible implementar la solución mediante un S.M.A.S utilizando una aproximación «servicio externo reactivo» tal y como se describe en el punto 3.12.1.3.*

En este caso, en lugar de un tablero tipo ajedrez vamos a plantear un entorno mucho menos discretizable. Para la ejecución de este experimento se requerían los siguientes elementos

- Apoyándonos en la **metáfora del entorno** vamos a recrear un entorno virtual que sea lo más parecido posible a un entorno agreste con elevaciones del terreno, límites irregulares, obstáculos parciales, etc.
- Diseñaremos un conjunto de herramientas «estigmérgicas» bio-inspiradas en el concepto de feromona digital que expandiremos más adelante
- Plantearemos dos situaciones experimentales: un hormiguero que ha de buscar comida en su entorno y una bandada de agentes voladores que es capaz de organizar su movimiento para esquivar elementos extraños. Ambas experiencias se describirán con más detalle más adelante.
- Un framework software que permitiese la implementación como un servicio externo reactivo
  - Se utilizó una versión inicial del framework «**SSR-Hub**»
  - Dentro del Hub se implementó el módulo particular para el implementar los servicios externos deliberativos reactivos para resolver problema indicado y se denominó «**Sprawl2**».

Desde el punto de vista de la presente tesis, en este experimento se ha trabajado y/o demostrado directamente sobre los siguientes objetivos: *G1, G2a, G2b, G2c, G6, G7x, G8, G9, G10, G11b*

#### 4.4.2 Estrategia Sprawl2: Feromonas, Campos y Tags

Este servicio Sprawl2 también ha sido construido como un módulo de servicio del framework **SSR-Hub** y el modelo «despliegue-representación». En comparación a Sprawl1, Sprawl2 se centra en plantear soluciones desde una perspectiva menos clásica. En concreto deja de lado la lógica del «situation calculus» y apuesta por un sistema mixto deliberativo-reactivo. Dicho sistema se basa enteramente en un nuevo concepto operativo creado específicamente para este proyecto: el «**seed**».

Un **seed** es una herramienta software que toma características de feromonas digitales y campos computacionales y partículas cargadas. También toma prestadas de la física los conceptos de dispersión, atenuación y forma de onda. En conjunto tendremos un «**campo de seeds**» puramente abstracto y al que se puede acceder únicamente a través de la capa de alto nivel (representación simbólica espacial) que será una representación colaborativa del espacio que detectan los agentes. Mediante las diferentes configuraciones de energía, forma de onda, atenuación, tiempo, etc. se puede conseguir infinitud de campos según nos interese. Pero lo más interesante es que a este concepto más o menos trillado se le ha añadido un «**tag**».

Un tag es un símbolo asociado a la información posicional, energética e intencional. Es decir, los agentes, además de percibir cierta energía que les indica la posición y distancia de un seed, perciben directamente el símbolo asociado a esos seeds. Como estos tags son definidos y parametrizados por los propios agentes, se convierten en el vehículo semántico de su intercambio de información.

A la hora de acceder a la información, los agentes pueden usar los «**tagargs**», una serie de parámetros acumulativos asociados a la consulta con una gran potencia expresiva que permite programar un agente sencillo en poco más de 15 líneas de código.

Para consolidar esta nueva herramienta se han realizado dos experimentos clásicos en los cuales los objetivos del problema son: probar el framework y la búsqueda de la inteligencia artificial en sí misma. Estas experiencias son:

- Búsqueda y almacenamiento de recursos (ant colony): Un conjunto de agentes explora en entorno espacial en busca de alimento para recolectarlo y almacenarlo.
- Bandadas dinámicas (flocking): Un conjunto de agentes se mueve libre y dinámicamente conformando una bandada compacta que es capaz de esquivar atacantes perdiendo momentáneamente la formación y volviendo a recuperarla automáticamente.

A continuación se describen en profundidad todos los detalles de los seeds, las funciones de propagación, los tags y los tagargs, así como los experimentos demostrativos.

#### 4.4.2.1 Antecedentes

El concepto de feromona digital aplicado a las colonias de termitas y hormigas (Dorigo et al., 2006) ya está completamente definido y asentado. De igual manera los campos asociados a los sistemas multiagente, como se puede observar en (Bandini et al., 2001, 2002; Mamei and Zambonelli, 2004), también se han utilizado en numerosas ocasiones.

En general se han investigado numerosos algoritmos y técnicas bio-inspiradas como optimización por colonias de hormigas, termitas, abejas, avispa, sistemas inmune-artificiales, organización de enjambre de partículas, etc. Sin embargo estas soluciones estaban acopladas a un determinado tipo de problema, ya fuese la optimización como en el caso de Dorigo en (Dorigo et al., 1996), (Dorigo and Gambardella, 1997) y (Dorigo et al., 2006), o a la búsqueda de algoritmos de navegación robótica como en (Arkin, 1989), siempre dichas herramientas eran parte de la solución a un problema dado.

En parte tiene bastante lógica que fuera así ya que no se tenía en cuenta el entorno como un concepto primordial en los sistemas multiagente. Desde esta nueva perspectiva del modelo despliegue-representación, tenemos un lugar donde «colocar» estos algoritmos fuera de los agentes propiamente dichos. Al extraer la responsabilidad de gestionar feromonas, campos, partículas, etc. de los agentes y traspasársela al entorno descargamos al agente de complejidad simplificando su funcionalidad.

#### 4.4.3 Descripción del concepto Seed

La palabra «seed» se construye como un acrónimo de «symbolic energy emission droplet», aunque también significa «semilla» en inglés. Un seed es un elemento intangible que conforma la representación de un espacio relativo a un entorno sobre el que un hub de agentes trabaja. Estos seeds tienen las siguientes características físicas:

- Un seed está asociado a una posición según un vector posicional  $\vec{P} \in \mathbb{R}^4$  que está definido en las cuatro dimensiones X, Y, Z, T del espacio asociado a un entorno en el nivel de despliegue. Las coordenadas X-Y-Z corresponden a la posición física donde reside ese seed en el espacio, mientras que la T (tiempo) corresponde al momento en el tiempo en el que ese seed fue generado.
- Un seed tiene una energía definida por el vector energía  $\vec{E} \in \mathbb{R}^4$  también definido en X, Y, Z, T. Este vector energía indica cuánta energía está propagando este seed

en cada una de las cuatro dimensiones. En la dimensión T expresa el diferencial energético que se aplica a las otras tres dimensiones espaciales.

- Para calcular diferencial energético en la creación del seed se almacena  $E_{T_0} \in \mathbb{R}$ , un valor escalar igual a la componente T del vector  $\vec{E}$  en el tiempo  $\vec{P}_T$  o lo que es lo mismo,  $E_{T_0} = f_p(\vec{E}_T, \vec{P}_T)$ . Siendo  $f_p$  la función de propagación que se describe más adelante.
- La función de propagación  $F_p$  viene definida por cuatro factores:
  - Una forma de onda asociada a la función de propagación definido por el vector  $\vec{Ws} \in \langle l, s, q, p, f \rangle$ . Cada una de las 4 dimensiones tratadas puede tener una forma de onda distinta. Estas son las posibles formas de onda: lineal(l), sinusoidal(s), cuadrada(q), polinomial de grado 1(p) y de un solo flanco(f). Se explican más adelante con detalle.
  - Un vector  $\vec{Wf} \in \mathbb{R}^4$  que define la frecuencia de la onda asociada a la función de propagación.
  - Un vector  $\vec{Wae} \in \mathbb{R}^4$  que define el exponente de atenuación en función de la distancia que se aplica a la onda asociada a la función de propagación.
  - Un vector  $\vec{Wd} \in \langle true, false \rangle$  que define sobre qué dimensiones se aplica el cálculo de distancia especial.
  - Y un valor discreto  $DM \in \langle e, m, c, k \rangle$  que define el método a seguir para calcular la distancia de este seed. Los posibles métodos son: Eculidea(e), Manhattan(m), Chebyshev(c), Minkowski(k). En este último caso también se indicará el parámetro p de la función de Minkowski.
- Por último, un seed puede ser de dos tipos distintos «persistente» o «atenuada». En el primer caso, el seed durará mientras que el sistema esté en marcha. En el segundo, el seed permanece mientras la energía que disipa en la dimensión T sea mayor que cero:  $f_e(\vec{E}_T, t_{now}) > 0$

Además tiene estas características simbólicas:

- Cada seed tiene un nombre. Dicho nombre puede repetirse o no y es definido a la hora de ser creado.



- También registra el UUID del agente generador de dicho seed. Un UUID es un identificador único que diferencia un agente de cualquier otro.
- Dispone de una lista de tags. Un tag es un elemento simbólico de tipo texto. Una lista de tags es un conjunto no ordenado de tags. Por ejemplo «tag1,tag2,tag3» es una lista de tags.
- Por último mantiene un vector intencional  $\vec{I} \in \mathbb{R}^4$ . Este vector almacena la información dinámica espacial del agente en el momento que se crea el seed.

El funcionamiento de un seed es simple: cuando es creado por un agente, dicho seed se almacena en el módulo «entorno simbólico aumentado» según los parámetros que ofrece el agente. La posición suele variar en función de la posición del agente, y los parámetros de energía y función de propagación suelen ser distintos según el problema que se está abordando. Así mismo los tags con los que se crea el seed agrupan de manera simbólica los seeds que comparten dicho tag.

#### 4.4.3.1 Función de propagación de la energía

El elemento clave de un seed es la función de propagación  $f_e : \mathbb{R}^4 \rightarrow \mathbb{R}^4$ . Dicha función es la que nos permite conocer la energía emitida por un solo seed en cualquier posición. Si tenemos un seed cuyo vector de posición es  $\vec{P}$  y deseamos averiguar la energía propagada hasta un punto  $\vec{O} \in \mathbb{R}$  en un tiempo determinado  $\vec{O}_T = t_1$  utilizamos la función de propagación. Sin embargo esta función no se comporta siempre igual ya que depende del vector  $\vec{W}s$  que puede tener cuatro valores en cualquiera de sus cuatro componentes dimensionales. Tomando como valores previos a la evaluación:

- Siendo  $D$  la dimensión a evaluar entre las posibles  $\langle x, y, z, t \rangle$
- Siendo  $\delta_{DM}(\vec{a}, \vec{b}, \vec{W}d)$  la distancia especial entre los puntos  $a$  y  $b$  según el método DM y el vector booleano de dimensiones afectadas.
- Siendo  $\alpha_p$  el coeficiente de propagación.

Tendremos que la función de propagación toma estos valores en función de la forma de onda:

- Si  $\vec{W}_{SD} = l$  (lineal):

$$f_{e_D}(\vec{P}, \vec{E}, \vec{O}, \vec{W}_{ae}, \vec{W}_f, \vec{W}_d, DM) = \vec{E}_D \cdot \alpha_p$$

- Si  $\vec{W}_{SD} = s$  (sinusoidal):

$$f_{e_D}(\vec{P}, \vec{E}, \vec{O}, \vec{W}_{ae}, \vec{W}_f, \vec{W}_d, DM) = \vec{E}_D \cdot \alpha_p \cdot \left[ \left[ \cos(\delta_{DM}(\vec{P}, \vec{O}) \cdot \vec{W}_f)_D + \Pi \right] / 2 \right] + 0,5$$

- Si  $\vec{W}_{SD} = q$  (cuadrada):

$$f_{e_D}(\vec{P}, \vec{E}, \vec{O}, \vec{W}_{ae}, \vec{W}_f, \vec{W}_d, DM) = \vec{E}_D \cdot \alpha_p \cdot \left[ |\delta_{DM}(\vec{P}, \vec{O}) \cdot \vec{W}_f)_D| \equiv 2 \right]$$

- Si  $\vec{W}_{SD} = p$  (polinomial grado 1):

$$f_{e_D}(\vec{P}, \vec{E}, \vec{O}, \vec{W}_{ae}, \vec{W}_f, \vec{W}_d, DM) = \vec{E}_D \cdot \alpha_p \cdot \left[ 1 - \delta_{DM}(\vec{P}, \vec{O}) \cdot \vec{W}_f)_D \right]$$

- Si  $\vec{W}_{SD} = s$  (un solo flanco):

$$f_{e_D}(\vec{P}, \vec{E}, \vec{O}, \vec{W}_{ae}, \vec{W}_f, \vec{W}_d, DM) = \begin{cases} \delta_{DM}(\vec{P}, \vec{O}) \leq \vec{W}_f)_D \rightarrow \vec{E}_D \cdot \alpha_p \\ \delta_{DM}(\vec{P}, \vec{O}) > \vec{W}_f)_D \rightarrow 0 \end{cases}$$

Por último necesitamos definir dos factores fundamentales, el coeficiente de propagación y la distancia:

#### 4.4.3.2 El coeficiente de propagación

Este concepto se basa en la «ley de la inversa del cuadrado» por la cual la energía se dispersa proporcionalmente a la inversa del cuadrado de la distancia y se argumenta su uso por ser un fenómeno común: «La ley de la inversa del cuadrado normalmente se aplica cuando alguna fuerza, energía, o cualquier otra cantidad conservada es emitida de manera radial en un espacio tridimensional desde un punto de origen» (Wikipedia). Sin embargo nos puede ser útil que este coeficiente no sea fijo al exponente 2 (cuadrado) como en la naturaleza, por lo que la ecuación cambia ligeramente dejando el exponente

parametrizable mediante el vector  $\vec{Wae}$  y en función de la dimensión que se está midiendo.

$$\alpha_p = \frac{1}{(\delta_{DM}(\vec{P}, \vec{O}) + 1)^{\vec{Wae}_D}}$$

#### 4.4.3.3 La distancia especial

El concepto distancia también es de sobra conocido, pero para esta herramienta vamos a utilizar la distancia especial. Cuando hablamos de distancia se sobreentiende que nos referimos a distancia Euclídea ya que es la más habitual y la que se da en la naturaleza de la física Newtoniana. Sin embargo existen otras métricas de distancia que nos pueden ser útiles a la hora de diseñar los campos de los seeds. Vamos a usar cuatro tipos distintos: Eculidea, Manhattan, Chebyshev y Minkowski todos en el espacio tridimensional limitado. Esta limitación se introduce mediante el vector booleano  $\vec{Wd}$  que indica sobre qué dimensiones hay que aplicar la medición de distancia y por lo tanto, sobre qué dimensiones hacemos la conversión de 3D a 2D o incluso a 1D.

- Si DM=e (Euclidea):

$$\delta_e(\vec{a}, \vec{b}, \vec{Wd}) = \sqrt{\sum_{i=1}^3 [(\vec{a}_i - \vec{b}_i)^2] \cdot \vec{Wd}_i}$$

- Si DM=m (Manhattan):

$$\delta_m(\vec{a}, \vec{b}, \vec{Wd}) = \sum_{i=1}^3 [\vec{a}_i - \vec{b}_i] \cdot \vec{Wd}_i$$

- Si DM=c (Chebyshev):

$$\delta_c(\vec{a}, \vec{b}, \vec{Wd}) = \max_{i=1}^3 ((\vec{a}_i - \vec{b}_i) \cdot \vec{Wd}_i)$$

- Si DM=k (Minkowski). Esta distancia es una generalización de la Euclídea para exponentes en función de p:

$$\delta_k(\vec{a}, \vec{b}, \vec{Wd}, p) = \sqrt[p]{\sum_{i=1}^3 [(\vec{a}_i - \vec{b}_i)^p] \cdot \vec{Wd}_i}$$

#### 4.4.3.4 Representación gráfica

Una vez se han declarado las ecuaciones que gobiernan la propagación de la energía en los seeds vamos a ver la representación gráfica de las distintas formas de onda de un seed.

$$\vec{W}s_D = l; \vec{E}_D = 10; \vec{W}a_e_D = 2 \rightarrow f(x) = 10 \cdot \frac{1}{(x+1)^2}$$

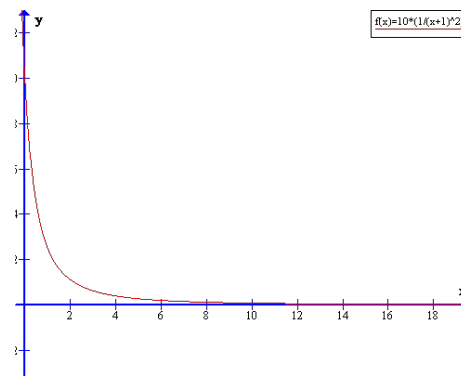


Figura 4.27: Propagación lineal con propagación cuadrática y energía inicial 10

$$\vec{W}s_D = s; \vec{E}_D = 10; \vec{W}a_e_D = 2; \vec{W}f_D = 5 \rightarrow f(x) = 10 \cdot \frac{1}{(x+1)^2} \cdot [\cos((x*5) + \pi) + 1]$$

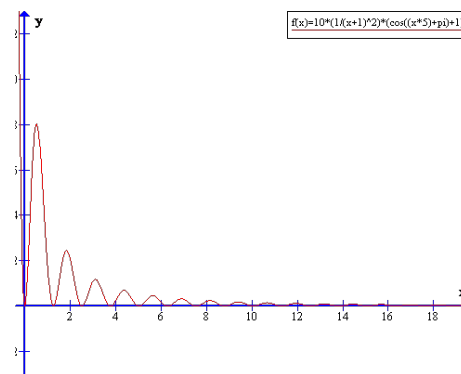


Figura 4.28: Sinusoidal con ate. cuadrática, frecuencia 5 y energía inicial 10

$$\vec{W}s_D = s; \vec{E}_D = 10; \vec{W}a_{e_D} = 2; \vec{W}f_D = 5$$

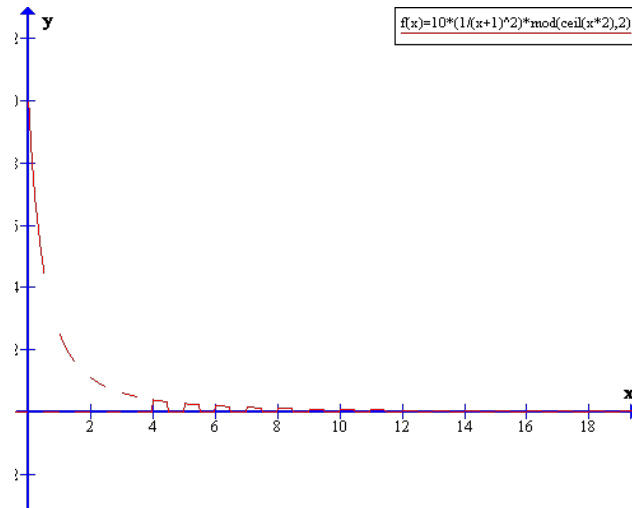


Figura 4.29: Propagación lineal con propagación cuadrática y energía inicial 10

$$\vec{W}s_D = s; \vec{E}_D = 10; \vec{W}a_{e_D} = 0; \vec{W}f_D = 1/10$$

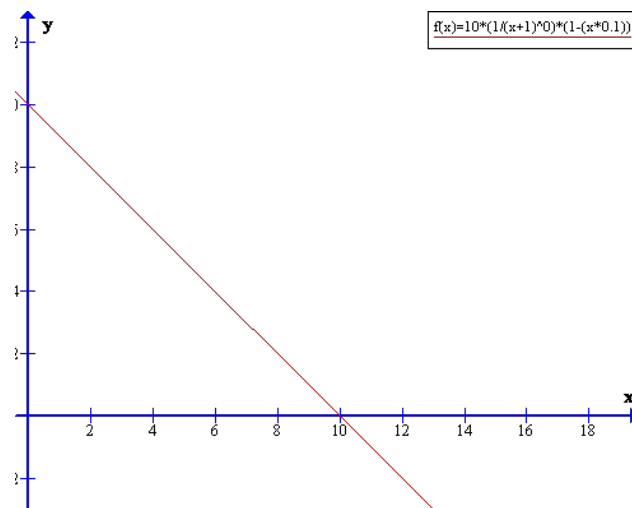


Figura 4.30: Propagación polinomial sin atenuación y frecuencia 1/10

$$\vec{W}s_D = s; \vec{E}_D = 10; \vec{W}a_{e_D} = 0; \vec{W}f_D = 10$$

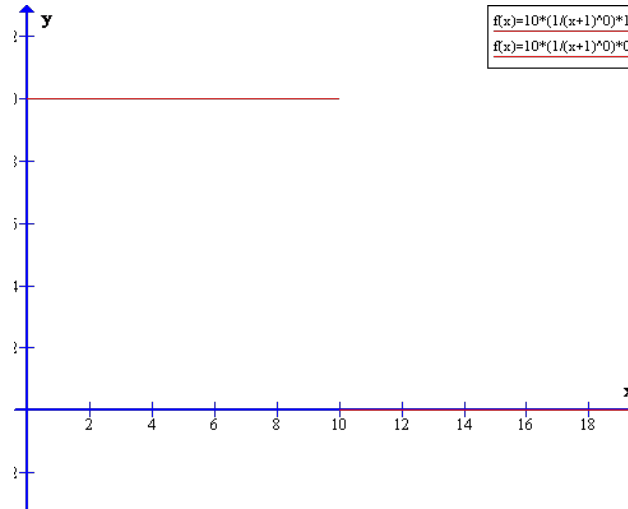


Figura 4.31: Propagación singleflank sin atenuación, frecuencia 10 y energía inicial 10

Pero para ver con mayor claridad los resultados de las anteriores funciones sobre un espacio 2D es mejor representar un espacio de 256x256 metros cuadrados realizando mediciones cada metro. Los tonos claros representan energía baja, mientras que los tonos oscuros representan energía alta. A continuación se muestran distintas configuraciones de seeds agrupados por forma de onda: lineales, sinusoidales, cuadradas, polinomiales y de un solo flanco.

**Forma de onda lineal** Como se observa en los diferentes ejemplos, el método de la distancia especial (DM), el exponente de atenuación y la limitación de dimensiones afecta en grado sumo a la manera en la cual la energía se transmite a lo largo y ancho del espacio.

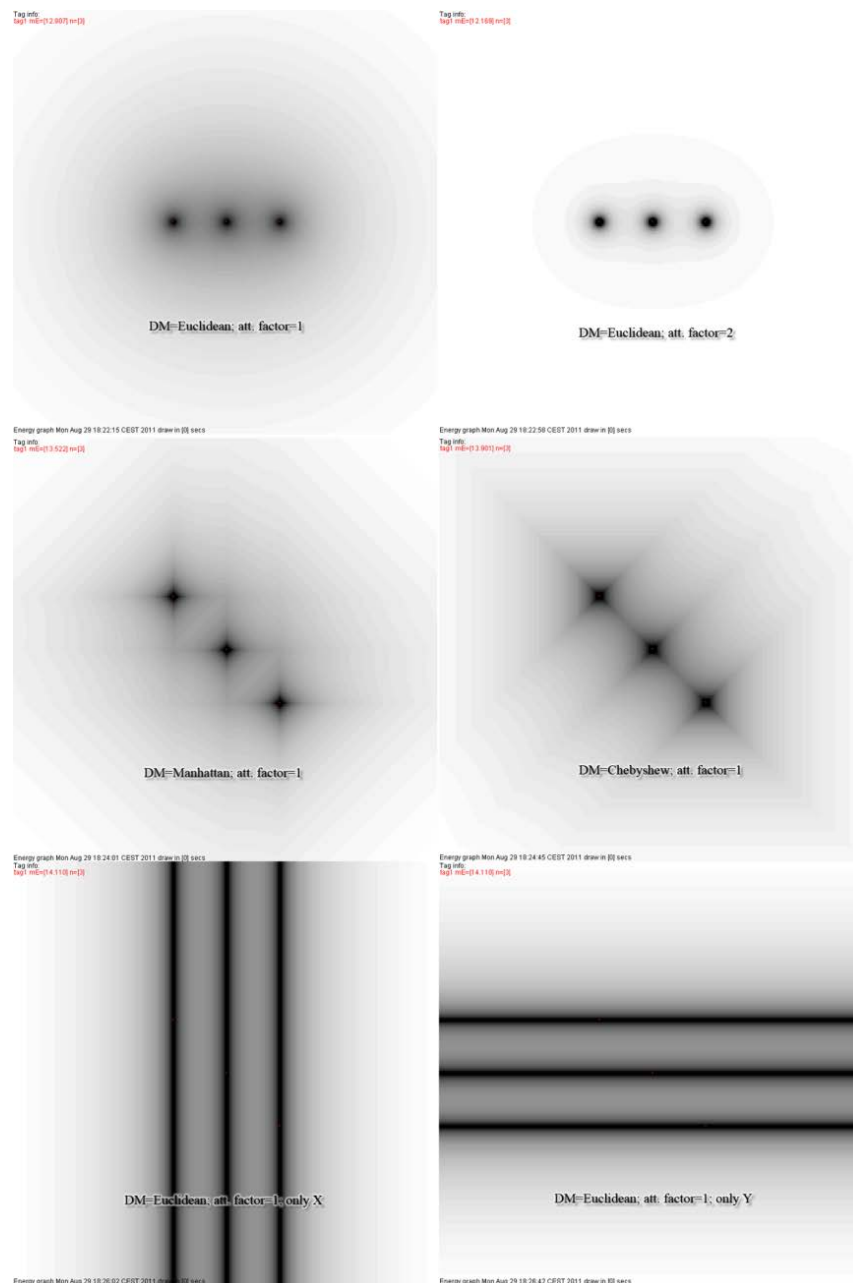


Figura 4.32: Ejemplos forma de onda lineal

**Forma de onda sinusoidal** Esta forma de onda, cuando se usa con la distancia Euclídea, se comporta como las ondas en la superficie de un embalse de agua. Esto es especialmente interesante para hacer que los agentes mantengan formación de manera automática.

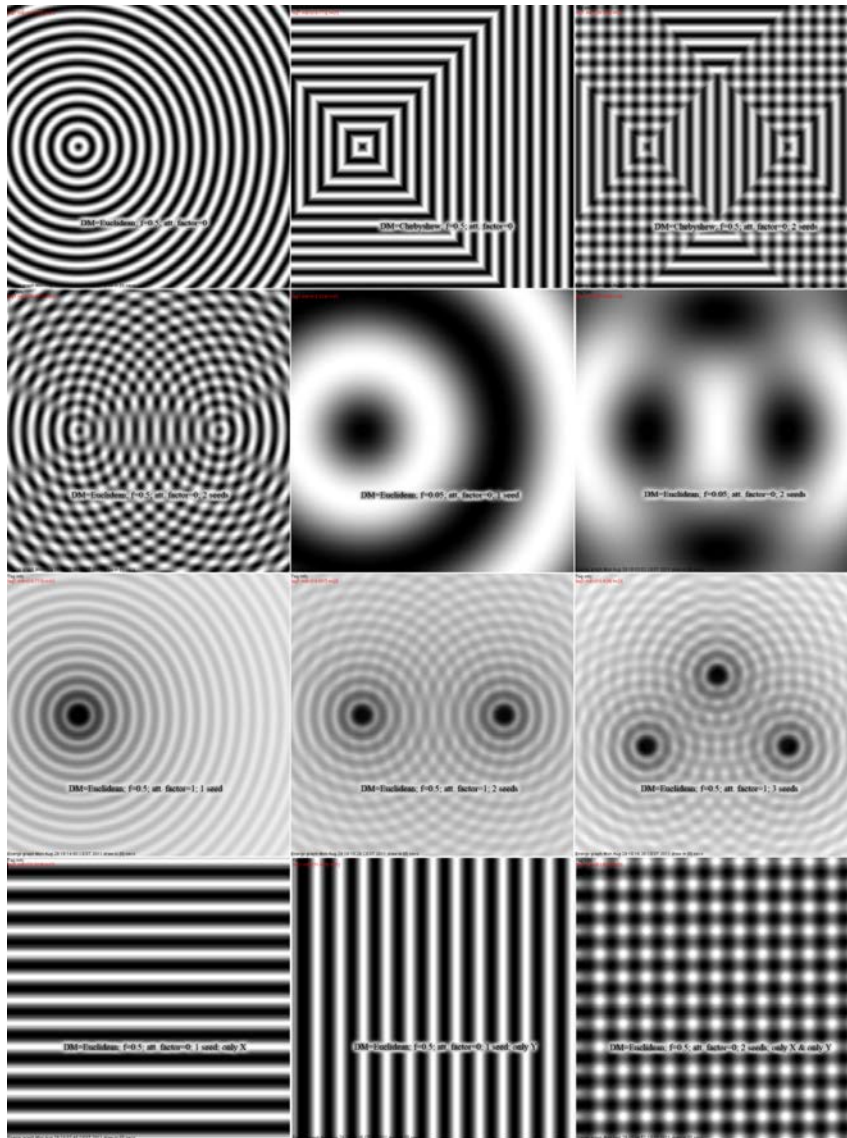


Figura 4.33: Ejemplos forma de onda sinusoidal



**Forma de onda cuadrada** Esta forma de onda se parece a la sinusoidal, pero se propaga por el espacio sin valores intermedios. Cuando se le aplica un factor de atenuación si existen valores intermedios, pero sólo del máximo.

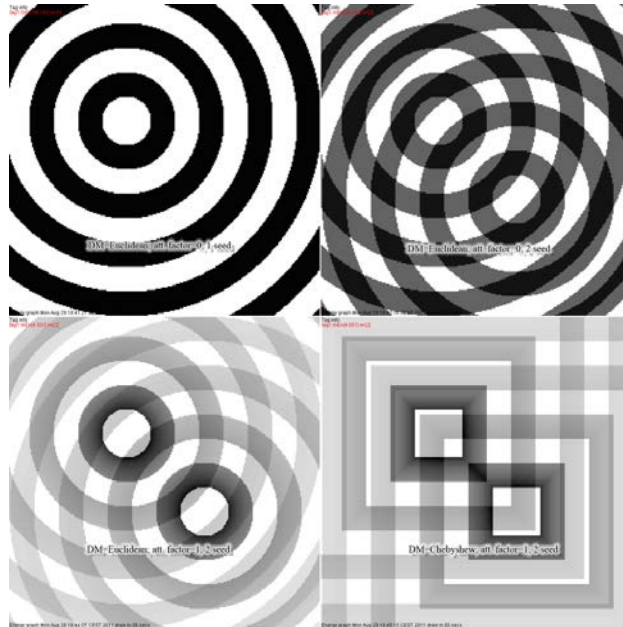


Figura 4.34: Ejemplos forma de onda cuadrada

**Forma de onda polinomial** En este caso se produce una dispersión lineal de los valores de la energía.

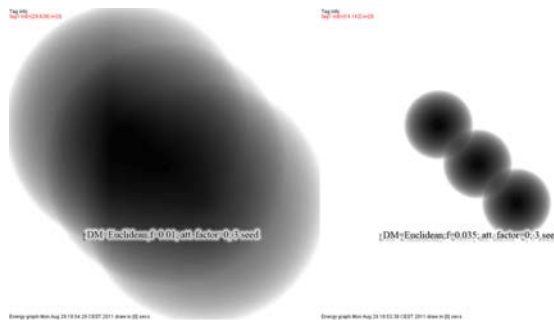


Figura 4.35: Ejemplos forma de onda polinomial

**Forma de onda un solo flanco** Puede que este sea la forma de onda más sencilla. En este caso, la energía se propaga linealmente hasta que llega al punto de flanco, donde deja de transmitirse bruscamente. Por supuesto si asociamos el coeficiente de propagación mayor de 0 sí que se producirá dicha atenuación, pero aun así el flanco determinará el final del campo. Esta es la fonda de onda más útil para definir zonas concretas de actuación y pueden ser fácilmente conjugadas con otras configuraciones (como se puede observar en el último gráfico) donde se ha incluido un típico campo de formación regular.

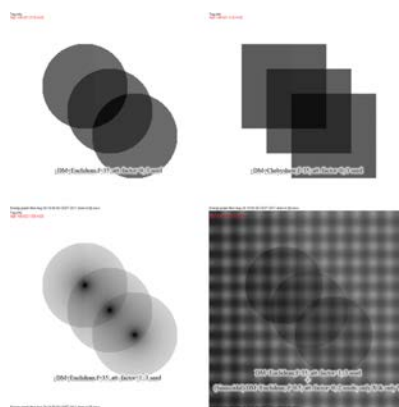


Figura 4.36: Ejemplos forma de onda un solo flanco

Las anteriores estas representaciones están realizadas en el plano 2D X-Y. Sin embargo los seeds emiten energía a lo largo de las tres dimensiones X-Y-Z, pero es muy complejo mostrar una medición espacial 3D a través de un medio 2D. Aun así se ha creado un tipo de agente especial llamado «puntero» que indica la dirección del ascenso del gradiente energético en su posición actual. Al poblar el espacio del entorno con una matriz 3D de dichos punteros conseguimos «intuir» el campo:



Figura 4.37: Matriz de punteros

#### 4.4.3.5 Campo de seeds (SeedSets) y medición de valores

El significado de tener una función de propagación es poder realizar mediciones en un punto diferente de  $\vec{P}$  (el lugar del espacio y el tiempo donde se ha creado). En el modelo matemático no existen límites a los valores de dicho vector, aunque como es lógico sí que existen «de hecho» al implementarlo. En cualquier caso vamos a definir el concepto «campo de seeds» independientemente de las dimensiones reales de dicho campo ya que para la herramienta un «campo de seeds» es simplemente un conjunto de seeds. El nombre interno que la herramienta da a esta clase es SeedSet y también es el nombre que usaremos en adelante para referirnos a este concepto.

Volviendo al modelo teórico, recordemos que podemos realizar una medición del campo energético que desprende un seed en base de su función de propagación .

Como ya hemos visto, dicha función tiene varios parámetros de entrada, por ejemplo:

$$f_{e_D}(\vec{P}, \vec{E}, \vec{O}, \vec{Wae}, \vec{Wf}, \vec{Wd}, DM) = \vec{E}_D \cdot \alpha_p$$

Sin embargo  $\vec{P}, \vec{E}, \vec{O}, \vec{Wae}, \vec{Wf}, \vec{Wd}$  y  $DM$  son valores propios del seed, con lo que vamos a trabajar con una fórmula simplificada de la ecuación eliminando los valores «internos» del seed y añadiendo el concepto de seed con el símbolo  $\mathbb{S}$  quedando de esta manera en el caso de la forma de onda lineal:

$$f_e(\mathbb{S}, \vec{O}) = \vec{E} \cdot \alpha_p$$

Recordemos que  $\vec{O}$  es el vector posicional del punto de observación desde el que se realiza la medición. Por lo tanto podemos afirmar que es posible realizar una medición sobre un seed  $\mathbb{S}$  dado desde un punto en el espacio-tiempo  $\vec{O}$  mediante la función  $f_e(\mathbb{S}, \vec{O})$  que devolverá un vector  $\vec{E}'$  en el espacio-tiempo con los valores de dicha medición.

Sin embargo un solo seed no tiene una carga simbólica útil por lo que debemos definir la manera en la que se podrán realizar mediciones a un SeedSet, sabiendo que es un conjunto de seeds. Esta operación se realiza mediante esta ecuación:

$$f_e(\text{SeedSet}, \vec{O}) = \sum_{i=1}^N f_e(\mathbb{S}_i, \vec{O}) = \vec{E}''$$

Por lo tanto para realizar una medición de energía a un SeedSet basta con sumar

linealmente todas las mediciones individuales que se realizan para cada una de los seeds que contiene el SeedSet.

Pero también podemos hacer dos tipos más de lecturas a un SeedSet además de la energía: direccional e intencional.

El vector direccional indica la dirección en la que se encuentra un seed  $\mathbb{S}$  desde un punto de observación  $\vec{O}$ . Para calcular dicho vector se utiliza la resta de vectores, sin embargo el módulo del vector resultante de dicha resta es mayor cuanto más lejos están el seed y el punto de observación, ya que indica la distancia entre ambos. Si utilizásemos esta medida tal cual, cuanto más lejos estén el seed y el punto de observación, mayor será el módulo de dicho vector, lo que parece anti-intuitivo. Por ello se ha definido el vector direccional mediante la siguiente ecuación:

$$f_d(\mathbb{S}, \vec{O}) = \left[ \frac{\vec{P} - \vec{O}}{|\vec{P} - \vec{O}|} \right] \cdot |f_p(\mathbb{S}, \vec{O})| = \vec{P}'$$

En la primera parte de la ecuación se haya el vector unitario de la resta de vectores, para luego ponderarlo con el módulo del vector de la medición de energía en el mismo punto. De esta manera independizamos la posición de la distancia y lo hacemos proporcionalmente mayor cuanto más energía se detecte de dicho seed en esa posición. En otras palabras, intercambiamos la distancia por la energía.

De manera análoga a la medición de energía sobre un SeedSet es posible hacer una medición direccional sobre todo el conjunto de seeds mediante esta ecuación:

$$f_e(\text{SeedSet}, \vec{O}) = \sum_{i=1}^N f_d(\mathbb{S}_i, \vec{O}) = \vec{P}''$$

O lo que es lo mismo, la suma directa de todos los vectores direccionales. Esto devolverá un vector cuyo módulo será el factor atrayente y su dirección el centroide de todos los seeds que se han medido.

De una manera muy similar se trabaja sobre los vectores intencionales. El vector intencional  $\vec{I}$  viene definido en el seed por el propio agente en el momento de ser creado. Representa la intencionalidad del espacio, por ejemplo indicando una dirección que se tomó desde ese punto, al estilo de una indirección de un lenguaje de programación o más tangiblemente, un cartel con una flecha. Podríamos decir que es semántica espacial. Las

lecturas sobre este vector se realizan mediante las ecuaciones:

$$f_d(\mathbb{S}, \vec{O}) = \left[ \frac{\vec{I} - \vec{O}}{|\vec{I} - \vec{O}|} \right] \cdot |f_p(\mathbb{S}, \vec{O})| = \vec{I}'$$

$$f_e(\text{SeedSet}, \vec{O}) = \sum_{i=1}^N f_i(\mathbb{S}_i, \vec{O}) = \vec{I}''$$

En resumen, los seeds (y por extensión los seedsets por adición directa) permiten realizar mediciones mediante las funciones  $f_d$ ,  $f_e$ ,  $f_i$  (energía, dirección e intención).

#### 4.4.3.6 Tags y tagargs

Como se definió anteriormente, cada seed almacena uno o más tags de información simbólica. Un tag es simplemente una cadena de caracteres que elige el agente que crea el seed en el momento de hacerlo. El objetivo de esta característica de los seeds es poder agruparlos o filtrarlos en función de cierta información simbólica arbitraria.

Por ejemplo, un agente recolector que detecta su objetivo en una determinada posición espacial puede crear un seed con el tag «comida», «X», o «3212» eso es transparente para el seed, ya que el contenido semántico de dicho tag debe ser aplicado por el propio agente y por extensión a sus «hermanos» del Hub de agentes. Obvia decir que otro agente que no comparta la información semántica de ese tag no podrá actuar en consecuencia, al igual que una persona que no habla un idioma desconoce el significado de un cartel escrito en dicho idioma.

Pero, una vez que un hipotético agente ha generado un seed con una carga semántica concreta ¿cómo puede otro agente aprovecharse de la misma? A través de los «tagargs».

Como vimos en el punto anterior es posible realizar tres distintas mediciones a un SeedSet: de energía, de dirección y de intención. Aunque lo veremos más adelante en profundidad, es prácticamente obvio que un Hub de agentes va a compartir un SeedSet de la misma manera que los primitivos sistemas multiagente utilizaban una pizarra para transmitirse información. Vamos a llamar simplemente «campo» al SeedSet de un Hub compartido. Todos los agentes podrán crear seeds en el campo y también podrán realizar mediciones sobre el mismo. Las mediciones sobre un SeedSet (y por extensión sobre el campo de un Hub) devuelven siempre un vector. En el caso más simple en el que sólo hay un tag en todo el campo el agente podría tomar ese vector, aplicar algún factor sobre él,

y utilizarlo para seguir su movimiento. Por ejemplo, si cada seed significa «peligro», el agente solicitará el vector direccional, lo multiplicará por -1 y lo utilizará para alejarse de dicho peligro. Sin embargo en ese escenario la semántica del seed debe ser interpretada completamente por el agente. Por otro lado ¿para qué queremos dotar de carga simbólica a los seeds si luego vamos a mezclarlos todos al realizar las mediciones?

Para solucionar este problema se utiliza el concepto «tagargs». Este término es una contracción de «tag» y «arguments» (argumentos en inglés), y sirve para filtrar el conjunto de seeds sobre las que se va a realizar una determinada medición. Cada vez que un agente solicita una medición al campo ha de indicar los tagargs asociados según esta sintaxis:

- $\langle \text{tagargs} \rangle ::= \langle \text{tagarg} \rangle [ ';' \text{tagarg} ] \dots [ ';' \text{tagarg} ]$
- $\langle \text{tagarg} \rangle ::= \langle \text{tag} \rangle , ' \langle \text{owner} \rangle ' , ' \langle \text{domain} \rangle ' , ' \langle \text{measure} \rangle ' , ' \langle \text{weight} \rangle$
- $\langle \text{tag} \rangle ::= '*' \mid (\text{string})$
- $\langle \text{owner} \rangle ::= \text{'any'} \mid \text{'own'} \mid \text{'others'}$
- $\langle \text{domain} \rangle ::= \text{'both'} \mid \text{'environment'} \mid \text{'local'}$
- $\langle \text{measure} \rangle ::= \text{'energy'} \mid \text{'direction'} \mid \text{'intention'}$
- $\langle \text{weight} \rangle ::= (\text{double})$

Cada elemento tiene los siguientes significados:

- **tag** → El nombre del tag
- **owner** → Filtra los seeds en función de su creador. Sólo puede elegirse entre cualquiera, el propio agente u otro agente distinto de si mismo
- **domain** → Como se verá más adelante, cada agente tiene acceso a dos SeedSets: uno el compartido con los demás agentes del hub, el «campo» y otro local propio que no comparte con nadie. Las mediciones pueden realizarse en uno o en otro dominio, o en ambos a la vez.
- **measure** → indica la medición a realizar

- **weight** → el peso que se asocia a ese tagarg y por el que se multiplicará el resultado.

Por ejemplo unos tagargs correctos serían definidos por esta cadena de caracteres:

*«food,any,env,intention,5;fire,any,env,direction,-10;body,others,env,energy,-1»*

Cuyo significado se puede resumir en tres partes:

- *food,intention,any,5* → busca seeds con tag «food», de cualquier agente, calcula el vector intencional y multiplícalo por 5
- *fire,any,direction,-10* → busca seeds con tag «fire», de cualquier agente, calcula el vector direccional y multiplícalo por -10
- *body,others,energy,-1* → busca los seeds con tag «body», de otros agentes, calcula la energía y multiplícalo por -1

Una vez calculados los tres vectores, se suman y se devuelve al agente. De esta manera, con una simple expresión el agente define los tags que le interesan y qué comportamiento tiene hacia ellos indicado por el campo weight. Otros factores semánticos como la definición del subconjunto «owner» ayudan a dar más potencia a esta herramienta.

También se podrían incluir otros filtros como distancia mínima del seed a detectar, dirección, etc. pero esos factores deberían ser controlados con los parámetros de creación de los seeds (energía, forma de onda, etc.).

#### 4.4.3.7 Diseño y funcionamiento del Hub

En el punto anterior se ha hecho referencia en varias ocasiones al **HUB** de agentes. Recordemos que el concepto de **HUB** de agentes se desprende directamente del modelo «despliegue-representación», más concretamente de la parte de «representación espacial simbólica». Es en dicho módulo donde se manejan las funciones de alto nivel de abstracción de los agentes, como es el caso de los seeds y los SeedSets.

En el modelo que se ha implementado en Sprawl2, el SeedSet común del **HUB**, al que se denomina simplemente «campo» se encuentra dentro del módulo de representación simbólica espacial, al acceso compartido de todos los agentes simultáneamente.

Pero en Sprawl2 cada agente tiene acceso a dos SeedSets: uno el «campo» compartido y el otro es un SeedSet local para cada agente. Este SeedSet local se encuentra en el módulo de gestión de Agentes.

Se ha llegado a esta decisión para dar flexibilidad a la herramienta. Aunque en los sistemas colaborativos es raro que un agente se guarde información para sí mismo, sí que es posible que en un entorno competitivo no sea así. El objetivo es dar a cada agente de un espacio privado donde definir sus propias construcciones cognitivas sin ser interferidas por los demás si así lo desea.

Este modelo permite realizar, además, volcados de información en ambas direcciones: si el agente así lo indica, todos los seeds del «campo» se volcarán en su propio SeedSet, y complementariamente, a una señal del agente, todo el contenido de su SeedSet se copiará dentro del «campo». Esta herramienta es útil, por ejemplo, si el agente de manera autónoma desea comprobar sus hipótesis antes de hacer pública su información. Por ejemplo, una hormiga exploradora podría evitar que sus feromonas sean visibles hasta el momento en el que encuentre comida. Obviamente este no es un comportamiento natural ya que esto es imposible de realizar en la naturaleza, pero resulta ciertamente útil.

Dentro del Hub también se encuentra el módulo de mensajería externa, que es el encargado de escuchar las solicitudes de los agentes y cursarlas bien al gestor de agentes o al entorno simbólico virtual. Estas son los posibles mensajes que es capaz de atender:

- **AddAgent():void** → El agente se añade al Hub. Este mensaje es trivial en Sprawl2 ya que siempre se acepta.
- **AddSeed(domain, seed):void** → El agente añade un seed con las características indicadas al SeedSet indicado en domain.
- **GetMeasure(observationPoint, tagargs):vector** → El agente solicita una medición del campo desde observationPoint con los tagargs indicados.
- **GetTracking(observationPoint, tagargs):vector** → El agente solicita un vector de dirección derivado de su actual vector de movimiento. El módulo gestor de agentes mantiene información de las últimas posiciones de cada agente para así poder establecer un vector de movimiento. En función de dicho vector y de su velocidad se establece una zona sensible o no. Por ejemplo si el agente se dirige al norte rápidamente detectará con mayor intensidad los seeds que se encuentran al



norte, de manera indistinta al este y al oeste, y muy atenuados los que están hacia el sur.

- **GetGradient(observationPoint, tagargs, up\_or\_down):vector** → El agente solicita un vector de dirección que ascienda o descienda el gradiente según se indique en el parámetro up\_or\_down. El módulo de representación espacial simbólica toma 27 mediciones alrededor del agente en todas las direcciones espaciales tridimensionales. Con ello construye una «maya» y compara las mediciones de los tagargs de cada uno de los 27 puntos. El agente indica que quiere ascender se busca el punto de esos 27 que más potencia genera y se le indica el vector direccional para llegar al mismo. Obviamente, si el agente desea descender el gradiente se elige el punto con menos potencial.
- **GetTags(domain, observationPoint):string** → El agente interroga al SeedSet del dominio indicado para que le devuelva una lista con todos los tags distintos que tienen los seeds. Dicha lista se ordena de mayor a menor energía medida en el observationPoint. Esta función es extremadamente interesante para los agentes BDI que usen Sprawl2 ya que permite «explorar» los símbolos del entorno. Sería factible, por ejemplo, un agente que investigue los tags desconocidos modificando los parámetros de los tagargs con aprendizaje por refuerzo o algoritmos genéticos desarrollando así un comportamiento simbólico autónomo.

Por último, el módulo de interfaz de usuario genera un mapa automático de los seeds y los campos energéticos que genera. Sin embargo este mapa es 2D sobre el plano X-Y por lo que la información que se ofrece es bastante limitada. En cualquier caso se han desarrollado agentes especiales sin otra función que indicar el estado del campo para poder dar visibilidad a dichos elementos ya que desde dentro del simulador no es posible percibir dichos campos más que con las acciones de los agentes. Este es un ejemplo:



Figura 4.38: Ejemplo gráfico 2D automatico

El diagrama de representación simbólica espacial del modelo personalizado para Sprawl2 queda, por lo tanto, como sigue:

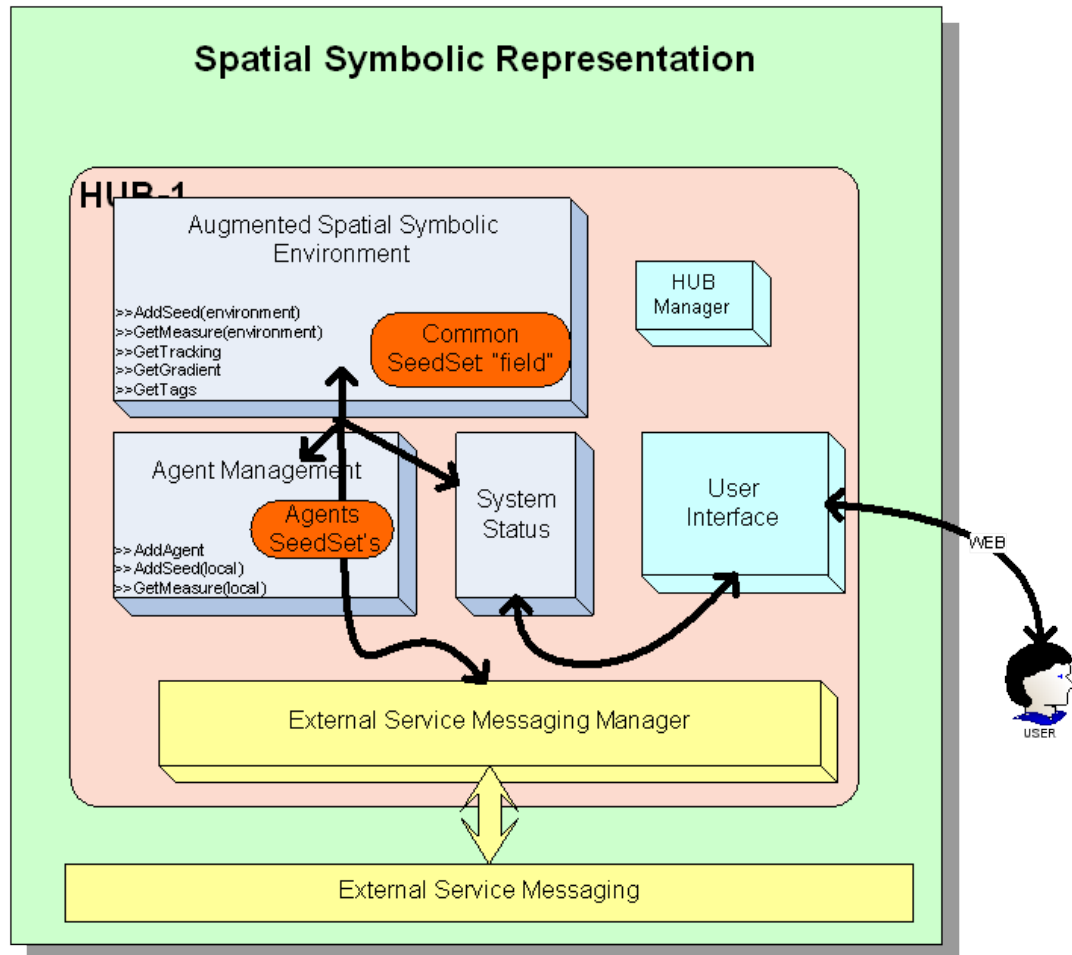


Figura 4.39: Representación para Sprawl2

#### 4.4.4 Experimento 1: Búsqueda y almacenamiento de recursos (ant colony)

Para poder comprobar la hipótesis de partida y también probar empíricamente la validez del modelo «despliegue-representación» y del framework **SSR-Hub** me planteo implementar un sistema multiagente situado que sea capaz de realizar una exploración en busca de un determinado elemento del espacio y su transporte y recolección.

##### 4.4.4.1 Enunciado del problema

Planteamos un problema abstracto con el siguiente enunciado:

*Se desea comprobar la viabilidad de un sistema multiagente que, utilizando únicamente Sprawl2 sea capaz de imitar el comportamiento de una colonia de hormigas: saldrán de un hormiguero único y explorarán el entorno en busca de alimento. Una vez encontrado, las hormigas deberán llevarlo de vuelta al hormiguero para su acumulación. Para acotar el experimento se define una cantidad fija de alimento en el escenario y se mide el tiempo que tardan un conjunto con  $N$  hormigas iniciales en recolectar toda la comida. La medida de éxito se realizará en función del menor tiempo empleado. Para hacer más interesante el problema existirán cuatro distintas fuentes de comida separadas en el espacio. El entorno espacial estará acotado para asegurar la convergencia del problema en el tiempo. Todo el experimento se realizará en **Open Simulator**.*

##### 4.4.4.2 Estructura de Despliegue

Para implementar este problema, aparte de disponer de Sprawl2 y sus seeds, hay que realizar cierto trabajo en la estructura de despliegue, dentro del proveedor de entorno (**Open Simulator**):

- Construir un recinto cerrado donde se situará el entorno del experimento (arquitectura digital).
- Crear los objetos «hormiguero» y «fuente de comida».
- Dotar a dichos objetos de inteligencia, programando los scripts en LSL necesarios para que detecten el contacto con las hormigas y almacenen o suelten comida

según les corresponda. Además el hormiguero deberá contabilizar el tiempo que ha pasado hasta que está lleno (ya tenga toda la comida del entorno), para poder cuantificar la medida del éxito.

- Y lo más importante, crear los agentes «hormigas».

**Perímetro del experimento** Se ha diseñado un perímetro irregular con paredes de un metro de alto que son suficientes para que las hormigas no puedan saltar sobre él.



Figura 4.40: Vista superior del recinto

Como se puede observar en las imágenes, se han añadido algunos árboles para darle mayor realismo. Aunque de cara a las hormigas no existe ninguna ventaja por tener árboles o estanques, hay que recordar siempre la metáfora del entorno y el principio de máxima verosimilitud: si queremos que este entorno tenga validez debemos darle un contexto lo más antropomórfico posible.

**Hormiguero y fuentes de comida** Se han diseñado como superficies que salen apenas unos centímetros del suelo para que detecten cuándo una hormiga está sobre ellos. En ese momento, en función de su naturaleza, le solicitan a la hormiga en cuestión que acepte o que suelte una unidad de comida. Esta comunicación se realiza por los canales propios del simulador y no salen de la capa de despliegue, por lo cual no tienen nada que ver con Sprawl2.



Figura 4.41: Interior del recinto

El hormiguero, además, es capaz de contar el tiempo desde que se le inicializa tocándolo mediante un avatar y cuando el número de unidades de comida almacenadas llega al máximo. Ese tiempo indicará cómo de buena es la solución con las hormigas agente que hayan trabajado.

**Hormigas-agente** Las hormigas-agente son el elemento más complejo de todo el entorno. En primer lugar tienen dos comportamientos simultáneos gestionados por dos scripts LSL en paralelo:

- El diálogo con el Hub de agentes de Sprawl2
- La interacción con el entorno

En el diálogo con el Hub, el agente, cuando es generado, lo primero que hace es solicitar conexión con el Hub. Cuando este le responde correctamente queda a la espera de las indicaciones del otro script.

El script de interacción es en el que se lleva toda la carga lógica de bajo nivel. Su comportamiento se basa en este simple autómata de estados:

En otras palabras: el agente comienza buscando comida, hasta que la encuentra, en ese momento busca su hormiguero hasta que lo encuentra y vuelta a empezar.

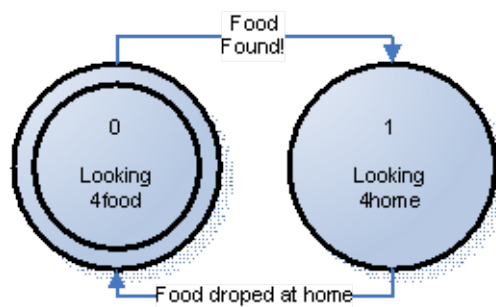


Figura 4.42: Autómata de estados de los agentes-hormiga

Lo que hace que el sistema funcione es que en cada estado se generan seeds distintos y se realizan mediciones distintas:

- Estado «looking4food»
  - Se generan seeds con:
    - \* Tag «4food»
    - \* Forma de onda lineal
    - \* Energía= $40 \cdot t \cdot 0.98^t$
    - \* Coeficiente de propagación=2
    - \* Tiempo de vida=5 minutos (el tiempo de vida se consigue definiendo la forma de onda del tiempo como singleflank y dándole valor  $300=5 \cdot 60$ )
  - Se usan los tagars:
    - \* 4food,own,env,dir,-1
    - \* 4home,any,env,dir,20
- Estado «looking4home»
  - Se generan seeds con:
    - \* Tag «4home»
    - \* Forma de onda lineal
    - \* Energía= $40 \cdot t \cdot 0.75^t$
    - \* Coeficiente de propagación=2

- \* Tiempo de vida=5 minutos
- Se usan los tagargs:
  - \* 4food,own,env,dir,20
  - \* 4home,own,env,dir,-1
  - \* 4home,others,env,dir,2

Interpretando los tagargs:

- Looking4food → le repelen sus propios seeds 4food, actuando como motor para que la hormiga no se quede nunca quieta y explore, mientras que también le atraen los seeds que cualquier hormiga emite cuando vuelve al hormiguero con comida
- Looking4home → atraen fuertemente los seeds que desprenden las hormigas que marchan hacia el hormiguero, mientras que sus propios seeds 4home le repelen y 4home de las otras hormigas le atraen levemente.

En cada ciclo de reloj de cada hormiga ésta le pregunta a Sprawl2 por el mejor camino a seguir haciendo tracking según los tagargs de cada estado. Una vez llega la respuesta, la hormiga se mueve en la dirección indicada.

#### 4.4.4.3 Resultados

Una vez construida toda la estructura de despliegue, y tras las lógicas pruebas básicas se lanzaron varias versiones del experimento. En las primeras versiones el carácter determinista del simulador hacía que siempre se repitiesen los mismos patrones. Para evitarlo se añadió una pequeña aleatoriedad en la posición de los Seeds  $\pm 0.05$  metros en X e Y. Con este pequeño cambio cada experimento es distinto.

Dado que el objetivo del problema no era la optimización del sistema, sino comprobar su viabilidad no se ha producido una experimentación profunda de los parámetros de los seeds y los tagargs que optimizan el tiempo de recolección. Sin embargo este problema podría ser estudiado en el futuro con más tiempo y un enfoque más general.

Estos son algunos recursos gráficos con los experimentos:

[http://www.youtube.com/watch?v=0um\\_oIHGY38](http://www.youtube.com/watch?v=0um_oIHGY38)

<http://www.youtube.com/watch?v=2SjIcxuBr1I>



### 4.4.5 Experimento 2: Bandadas dinámicas (flocking)

De nuevo con el objetivo de comprobar la viabilidad del sistema plantee un problema bastante conocido y simple: la organización espacial automática imitando el comportamiento de las aves o los peces, comúnmente llamado «flocking».

#### 4.4.5.1 Enunciado del problema

La consecución del problema responde al siguiente enunciado:

Se desea realizar una prueba de concepto en la que un número indeterminado de agentes «voladores» sea capaz de agruparse en una bandada cuidando mucho de no tocarse entre ellos a la vez que se alejan y esquivan a cualquier avatar que se les acerque. Si no tienen ningún avatar cerca la bandada deberá permanecer más o menos estática en su posición. El éxito de este problema es complejo de evaluar cuantitativamente, así que deberá ser comprobado de manera cualitativa.

#### 4.4.5.2 Estructura de Despliegue

En esta ocasión la estructura de despliegue es más sencilla, ya que no se especifica ningún límite al espacio. Por lo tanto únicamente debemos construir los agentes voladores.

Al igual que en el experimento de las hormigas, los agentes tendrán dos partes:

- El diálogo con el Hub de agentes de Sprawl2
- La interacción con el entorno

La parte de interacción tendrá un solo estado pues su comportamiento es muy simple. Por cada ciclo de reloj de cada agente:

- Se genera un seed «flock» en la posición del agente
- Si detecta un avatar genera un seed «alien» en la posición del avatar
- Solicita la dirección de incremento del gradiente y se mueve en consecuencia de los tagargs

Por lo tanto definimos ya la configuración de seeds que utilizarán para organizar su Hub.

- Se generan seeds con:
  - Tag «flock»
  - Forma de onda sinusoidal
  - Frecuencia=0.5
  - Energía=100
  - Coeficiente de propagación=2
  - Tiempo de vida=tiempo entre cada ciclo de reloj (1 segundo)
- Se generan seeds con:
  - Tag «alien»
  - Forma de onda lineal
  - Energía=1
  - Coeficiente de propagación=2
  - Tiempo de vida=tiempo entre cada ciclo de reloj (1 segundo)
- Se usan los tagars:
  - flock,other,env,energy,1
  - alien,any,env,dir,-1

Como es obvio, los seed «flock» les atraen y los seed «alien» les repelen. Sin embargo hay que tener en cuenta que el seed flock es de tipo sinusoidal, lo que hará que el espacio directamente anexo al agente (definido por la frecuencia) tenga un vector energía mucho menor que el que se encuentre a  $1/\text{frecuencia}$ , donde será máximo. De esta manera los agentes voladores mantienen un equilibrio entre todas sus posiciones relativas que solo es alterado por la presencia de seeds «alien» que les repelen de manera inversamente lineal.

#### 4.4.5.3 Resultados

El resultado de este experimento fue satisfactorio prácticamente desde el primer intento. Como se indica en el enunciado del problema es en extremo complicado hacer una evaluación cuantitativa del éxito de la solución, aunque de manera análoga al experimento de las hormigas, el objetivo no era tanto realizar un sistema afinado sino mostrar las posibilidades tanto del modelo teórico como de la herramienta Sprawl2.

En cualquier caso, para poder observar el comportamiento de estos agentes se puede consultar este recurso:

<http://www.youtube.com/watch?v=NL8-oRusZuI>

#### 4.4.6 Conclusiones

Tal y como se ha comentado en los apartados de resultados de los dos experimentos propuestos, el objetivo de los experimentos no era optimizar al máximo los parámetros de funcionamiento de los «tagargs» sino comprobar la viabilidad del modelo. Si recordamos la hipótesis de trabajo:

*[...] partiendo también de un **V.I.E** en el que se da un problema genérico de alta complejidad tanto deliberativo como reactivo, es posible implementar la solución mediante un **S.M.A.S** utilizando una aproximación «servicio externo reactivo» tal y como se describe en el punto 3.12.1.3.*

El objetivo es comprobar si es realmente viable la implementación de servicios de alto nivel de complejidad en **V.I.E's** mediante **S.M.A.S's** y un enfoque mixto deliberativo-reactivo, lo que queda de sobra probado. Por tanto la hipótesis queda comprobada e indirectamente también los siguientes objetivos: *G1, G2a, G2b, G2c, G6, G7x, G8, G9, G10, G11b*

## 4.5 Experimento OS/2016

### 4.5.1 Introducción

Cuando en 2011 se elaboró el programa de estudios del Master en Ciencia y Tecnología Informática de la Universidad Carlos III de Madrid, dentro de la asignatura **Modelado, Simulación y Optimización** (normalmente abreviado por **MSO**) se decidió proponer un experimento práctico a los alumnos que estuviese relacionado con técnicas de **simulación** dentro de *Entornos Virtuales Inmersivos*.

El principal objetivo académico de esta actividad es poner a los alumnos en contacto con estrategias poco comunes al afrontar problemas de alta complejidad. Al introducir a los estudiantes en un entorno inmersivo, que no es otra cosa que un gran simulador, conseguíamos un cambio de perspectiva: de ver los simuladores como una herramienta externa y compleja a vivir en primera persona la situación.

Como objetivo secundario se planteó utilizar las propuestas descritas en el capítulo 3 para diseñar, evaluar e implementar el *Entorno Virtual Inmersivo* para realizar la práctica docente. Este es un ejemplo excepcionalmente claro, completo, conciso y acotado de necesidad de construir un **V.I.E** para un uso real, por lo que constituye caso de uso perfecto para evaluar la mayor parte de las propuestas emitidas.

### 4.5.2 Planteamiento iterativo de la experiencia

Como se ha indicado anteriormente, esta experiencia comenzó en 2011 y se ha seguido utilizando anualmente hasta la fecha (2017).

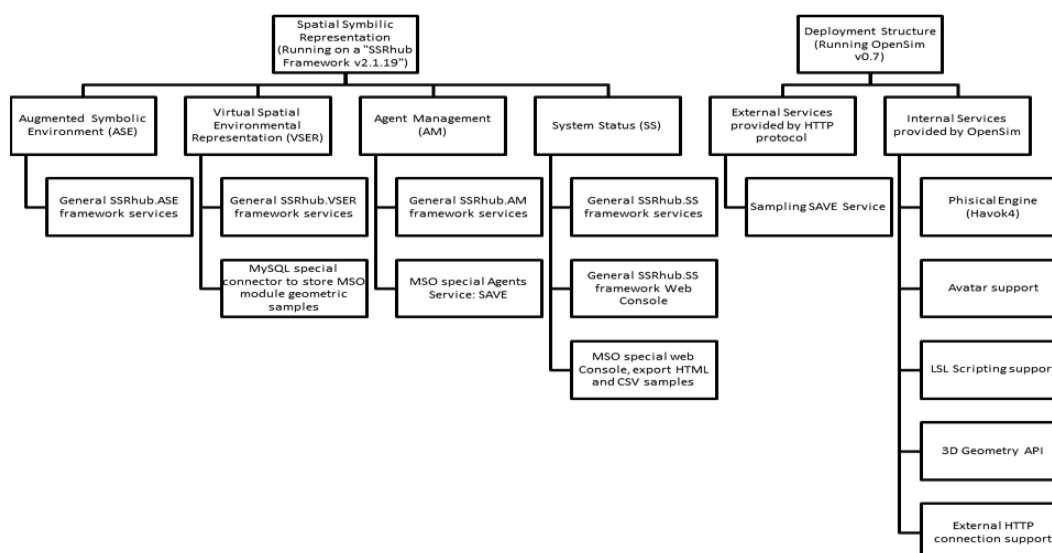
El hecho de ser una actividad académica iterativa con una periodicidad anual conforma un entorno perfecto para el perfeccionamiento gradual tanto de la experiencia de los alumnos como de las propias técnicas y métodos para construir el **V.I.E**.

Durante estos seis años se han intentado aproximaciones siempre desde la misma perspectiva (utilizar **V.I.E's**) pero se han ido variando los métodos y las herramientas siempre en aras de la optimización de la actividad docente. Algunas de las lecciones aprendidas / decisiones que se han ido tomando se describen en detalle en el punto 4.5.7 pero podemos adelantar que tras cinco sesiones de refinamiento, en 2015 se reconstruyó

el entorno desde cero aplicando la nueva metodología **PIIVE-DM**, la arquitectura **DRM** y el uso del framework **SSR-Hub**.

Adicionalmente, una publicación resumiendo esta experiencia y denominada «**A Propose Architecture for Situated Multi-agent Systems and Virtual Simulated Environments applied to Educational Immersive Experiences**» ha sido aceptada en el congreso **IWINAC2017 - 7th. INTERNATIONAL WORK-CONFERENCE on the INTERPLAY between NATURAL and ARTIFICIAL COMPUTATION** que se llevará a cabo del *19 al 23 de Junio de 2017 en A Coruña, España*.

Figura 4.43: Implementación concreta DRM en SSR-Hub para el proyecto MSO



Esta experiencia tiene un doble objetivo docente e investigador que se describen a continuación

### 4.5.3 Objetivo docente de la experiencia

Habitualmente las experiencias prácticas en materia de simulación se realizan a través de herramientas específicas tipo «CAD» como *Autodesk Simulation*, *CREO* o «*inhouse*», que aunque cumplen perfectamente su función tienen una baja recepción por parte del alumnado ya que tienen una curva de aprendizaje bastante elevada.

Detectado este **problema** en el ámbito docente, se planteó una hipótesis de trabajo: *si es posible plantear una experiencia práctica de simulación que presente un índice*

*ecológico más elevado podremos mejorar la recepción en el alumnado y minimizar el coste de adaptación a la herramienta.*

Para comprobar esta hipótesis se plantea el uso de un **V.I.E** con un índice ecológico  $I_{eco}$  medio que mejore la experiencia clásica. Esta premisa se sustenta sobre la metáfora del entorno 1.4.1 y por tanto en el resto de principios de diseño y formalización que se han propuesto en la presente tesis.

#### 4.5.4 Objetivo investigador de la experiencia

En paralelo vamos a utilizar esta experiencia como prueba de uso real de la metodología **PIIVE-DM** propuesta en la presente tesis. A continuación se muestra el resultado de la ejecución de los cuatro pasos de los que la metodología consta.

##### 4.5.4.1 Metodología PIIVE-DM

Este es el resumen de los pasos ejecutados de la metodología **PIIVE-DM**:

##### Paso 1: Definición de Requisitos

- ¿Qué?
  - Un **V.I.E** que permita a los alumnos de MSO realizar una práctica de simulación. Llamaremos a estos **V.I.E** *VIL* (Virtual Immersive Laboratory).
  - La práctica se detalla más adelante, pero se puede resumir como un registro de información espacial de cada agente del sistema. Los usuarios generarán tantos agentes como deseen y estos automáticamente almacenarán su información posicional en un **servidor**.
  - Posteriormente los alumnos podrán consultar ese **servidor** mediante un acceso web para poder recuperar los datos del experimento y analizarlos a través de un proceso externo.
  - Aunque se detallará en el paso 2 de esta metodología, toda la práctica se basa en un simulador de física newtoniana por lo que como requisito indispensable deberán implementarse los servicios físicos  $\Psi_{E,F}$

- ¿Dónde?

- En un **V.I.E** standalone *VIL* para evitar interferencias entre alumnos, aunque la información del resultado de la práctica deberá estar centralizado.
- En cuanto a la topología y aspecto del *VIL*, deberá ser lo más simple posible dentro de respeto a la **Metáfora del Entorno**.
- Deberá existir un entorno antropomórfico básico: un espacio abierto en un terreno (suelo) de tamaño indeterminado y condiciones atmosféricas realistas.
- Como la práctica va a consistir en someter a movimiento aleatorio a un conjunto de objetos-agente, caso de uso bastante corriente en cualquier laboratorio real de física o de química mediante el uso de un agitador se deberá dotar al *VIL* de un elemento que cumpla una función similar que denominaremos «batidora» o «agitadora» y cuya funcionalidad será únicamente mover a los objetos-agente que los alumnos introduzcan en el mismo.
- La «batidora», al ser un elemento móvil, deberá poder ser operada por el avatar de una manera sencilla e intuitiva, a ser posible mediante interacción táctil (sin recurrir a interfaz CLI o 2D).
- Dicha operación será un control para iniciar y otro para parar el movimiento de la «batidora».

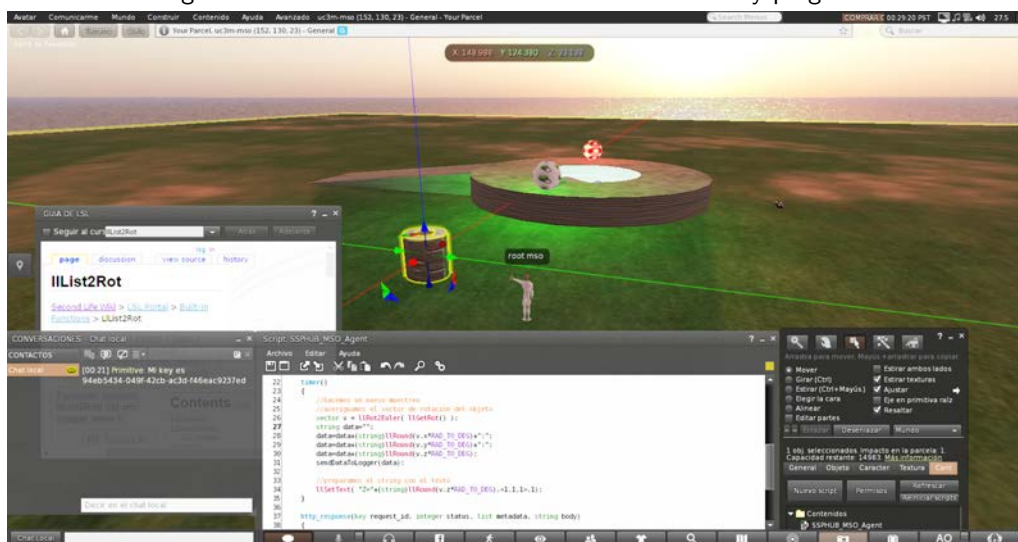
- ¿Quién?

- Los alumnos de MSO serán los únicos usuarios que podrán entrar en sus *VILs* personales.
- Por lo tanto los *VILs* deberán tener preconfigurado un avatar por defecto que los alumnos puedan usar sin tener que dedicarle esfuerzo a la creación de uno propio.
- Al estar los *VILs* desconectados unos de otros no hay problemas de seguridad ni de acceso concurrente y a todos los efectos serán monousuario.
- Todos los *VILs* de los alumnos deberán ser copias iguales (todos partirán de un **ObjectData**= $\Omega_0$ ), aunque a partir de su primera instanciación tendrán serán distintos (cada *VIL* de cada alumno tendrá su **ObjectData**= $\Omega_i$ ) con su propio entorno persistente.

- ¿Cómo?

- Será responsabilidad del cuerpo docente construir el  $VIL$  inicial, que se denominará  $VIL^{\Omega_0}$  con la «batidora» creada y operativa, el avatar por defecto generado, los scripts base necesarios para la práctica creados y compilados y un pequeño tutorial de instalación del  $VIL^{\Omega_0}$
- Analizando la estructura de despliegue **DS**, cuando los alumnos instancien un número indeterminado  $n$  de  $VIL^{\Omega_i}; i = 1, 2, 3 \dots n$ , existirán también  $n$  batidoras distintas agitando objetos, que a su vez estarán requiriendo los servicios de alto nivel de la capa **SSR**
- La capa de representación simbólica espacial **SSR** deberá tener una sola instancia y deberá ser ejecutada en las instalaciones de la Universidad para asegurar el depósito seguro y compartimentado de los datos experimentales.
- Los horarios en este caso son sencillos: El **SSR** deberá estar ofreciendo el servicio continuamente en horario 24x7, y los  $VIL$ s de los alumnos estarán dando el servicio bajo demanda, es decir, cuando los alumnos ejecuten localmente el **DS**

Figura 4.44: VIL: interfaz de usuario donde construir y programar



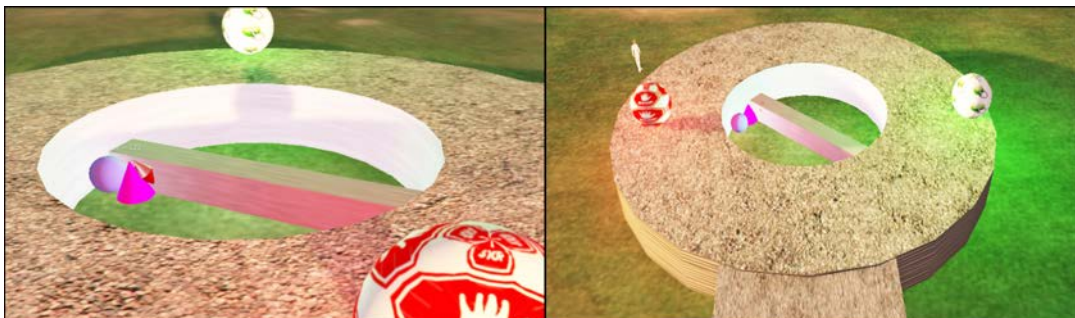
## Paso 2: Arquitectura de Servicios

- En este caso se requiere un entorno con unos *índices inmersivo y ecológico* moderados.



- Dado que se descarta el uso de hardware de inmersión (gafas o cascos) por motivos operativos, logísticos y económicos el interfaz físico se realizará mediante el modelo pantalla-teclado-ratón. Hoy en día (2017) disponer de un set **V.R** disponible para cada alumno en los laboratorios de la Universidad es inviable.
- Es decir, ya sabemos que  $I_\Phi$  (VIL) va a estar limitado por el hardware.
- Eso significa que el  $I_{eco}$  (VIL) se verá penalizado en los servicios sensoriales ( $\Psi_\Phi$ )
- Para compensar esta carencia la metodología recomienda potenciar el resto de servicios ecológicos  $\Psi_{eco}$ , es decir:
- Servicios ecológicos/físicos  $\Psi_{E.F}$ : debe existir un conjunto completo de servicios de física newtoniana tanto para aumentar el índice ecológico como para poder ejecutar la propia práctica ya que la «batidora» utilizará estos servicios.
- Servicios ecológicos/perceptivos/propioceptivos  $\Psi_{E.P.P}$ : deberá existir el concepto de avatar y tener apariencia antropomórfica
- Servicios ecológicos/interactivos  $\Psi_{E.I}$ : al tratarse de entornos monousuario no serán indispensables los servicios de habla e intercambio  $\Psi_{E.I.H}$  y  $\Psi_{E.I.I}$  pero el servicio táctil será requerido para operar la «batidora»  $\Psi_{E.I.T}$
- Todos estos servicios ecológicos serán ejecutados en cada uno de los  $VIL^{\Omega_i}; i = 1, 2, 3 \dots n$ , es decir, en la *estructura de despliegue*
- Servicios tecnológicos: Sólo se requerirá un servicio tecnológico: envío de información posicional de los objetos-agente a estudiar. Este servicio se implementará en la capa de representación simbólica espacial **SSR**

Figura 4.45: Detalle VIE-MSO: Dos vistas de la «batidora» en el VIL



### Paso 3: Estudio de Despliegue

- En este caso, y tras estudiar los *eco* necesarios se ha optado por utilizar un **Proveedor de Entorno** para dar todo el servicio de estructura de despliegue.
- El proveedor de entorno elegido es **OpenSimulator**.
- Esta plataforma software es capaz de ejecutarse en un equipo típico que pueda poseer el alumno (sólo requiere Microsoft.NET-Framework) y es capaz de paquetizarse en un solo archivo comprimido y redistribuirse de esta manera.

### Paso 4: Diseño de Tecnología

- El servicio tecnológico que se requiere desarrollar se denominará **MSO** y será capaz de recibir información de los objetos-agente que se están moviendo dentro de las batidoras y almacenarlo en una base de datos central
- Sin embargo para no tener que desarrollar todo un modelo **SSR** se va a utilizar el framework propuesto y utilizado en anteriores experimentos denominado **SSR-Hub**.
- Mediante la utilización de este framework desarrollado en *Java* y mediante el *IDE Eclipse* se ha implementará el servicio **MSO** dentro de los módulos **AM** y **VSER**

#### 4.5.4.2 OpenSim como Proveedor de Entorno

Como ya se ha descrito en el punto 3.10.2 *Open Simulator* es un software de libre distribución que ha evolucionado a partir de *Second Life* manteniendo la mayor parte de sus características y dando la posibilidad de ejecutar tantas instancias del simulador como se deseen, lo que lo hace óptimo para esta experiencia.

#### 4.5.5 Descripción de la experiencia práctica

Dado que el objetivo de la asignatura MSO no es mostrar a los alumnos la metodología **PIIVE-DM** ni la arquitectura **DRM**, en el enunciado de la práctica no se hace referencia a toda la teoría subyacente ya que se podría complicar y oscurecer la percepción de la experiencia, justo el objetivo contrario que tenemos.

El enunciado de la práctica queda, por tanto, de la siguiente manera:

*Una agencia espacial está estudiando diseños alternativos para robots exploratorios que se utilizarán en el reconocimiento de planetas de la Sistema Solar. Uno de estos diseños se basa en la premisa de usar una cubierta sólida lo más homogénea posible. Con este propósito, es necesario estudiar cómo evolucionan las diferentes formas geométricas tridimensionales en la orientación de sus ejes espaciales para situaciones de movimiento caótico. Por lo tanto, será necesario realizar un estudio estadístico de la evolución de los ángulos de los ejes X, Y y Z en el tiempo para una amplia gama de objetos con diferentes formas geométricas.*

*Para simplificar el alcance de esta sesión práctica limitaremos como mínimo y obligatorio el estudio únicamente del eje Z, y las formas geométricas a la de una esfera simple.*

#### 4.5.5.1 Descripción del experimento:

- **Hipótesis** una esfera perfecta mantiene equiprobabilidad en todos sus ejes.
- **Variables** decisión y respuestas: valores de los ejes
- **Eventos** a describir: el valor concreto de las componentes Z a lo largo del tiempo de manera discreta iterativa.

#### 4.5.5.2 Enriquecer la experiencia:

Una vez cumplidos los requisitos mínimos de la práctica, se valorará positivamente el enriquecimiento del modelo y las conclusiones obtenidas. Algunas sugerencias al respecto:

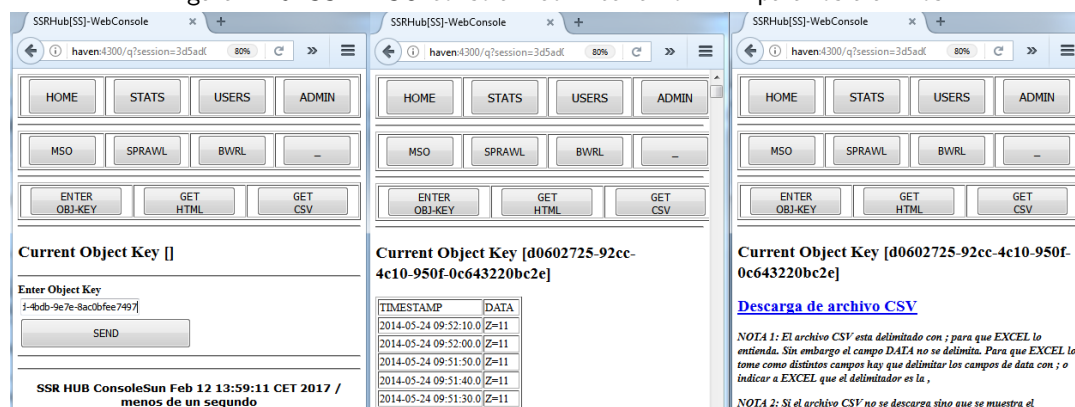
- **Medir**, registrar y analizar los componentes X e Y de los agentes.
- **Extender** los objetos que se experimentan: huso, toros, cilindros, etc.
- **Proponer** nuevos dispositivos o situaciones para generar movimiento aleatorio: agitadores dobles, pozos, espirales, etc.

### 4.5.5.3 Análisis de los datos

**Almacenamiento** , todas las muestras enviadas por los Agentes se almacenarán en un servidor **SSR-Hub** alojado en las instalaciones de la Universidad.

**Recuperación** , estas muestras pueden consultarse en cualquier momento a través de la Consola Web, tomando nota del UUID (código de identificador único universal) del Agente. Este UUID sólo es conocido por el propietario del Agente para garantizar la confidencialidad de los experimentos.

Figura 4.46: SSR-MSO consola web: interfaz off-VIL para los alumnos



**Conversión** , una vez que el alumno ha solicitado los datos experimentales en la Consola Web del **SSR-Hub**, se puede importar el conjunto de muestras de un Agente específico (en formato HTML o CSV) a otra herramienta (Excel, Matlab, etc.)

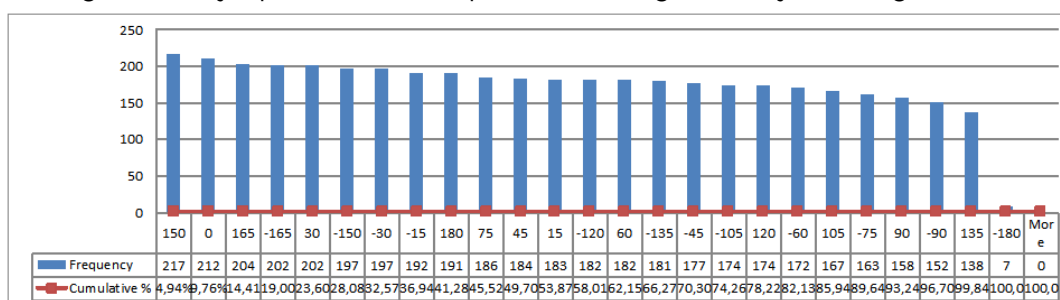
**Estudio** , a partir de ese momento, se utilizarán técnicas estadísticas estándar para analizar la información recibida (análisis de frecuencia, histogramas, gráficos, etc.) que será el resultado de la práctica, junto con las conclusiones sobre cumplimiento o no de la hipótesis de trabajo inicial.

## 4.5.6 Respuesta de los alumnos

El esfuerzo por parte del equipo docente para cambiar el enfoque sobre la herramienta de simulación clásica a una inmersiva ha sido muy bien recibida por los alumnos a lo largo de todo el ciclo de investigación.

Los estudiantes a menudo encuentran estimulante el uso de herramientas inusuales, especialmente cuando tienen un gran componente visual. Aunque en primera instancia este hecho puede «distraer» un poco la atención del estudiante, principalmente por su similitud con la estética clásica de juegos en tercera persona, rápidamente perciben el espíritu general de la actividad.

Figura 4.47: Ejemplo de resultado experimental: histograma de eje Z en ángulos de 15°



### 4.5.7 Lecciones aprendidas / toma de decisiones

Tal y como se ha indicado al comienzo de la descripción del experimento, la naturaleza recurrente de una asignatura es un campo de pruebas magnífico para conseguir una mejora iterativa de una solución.

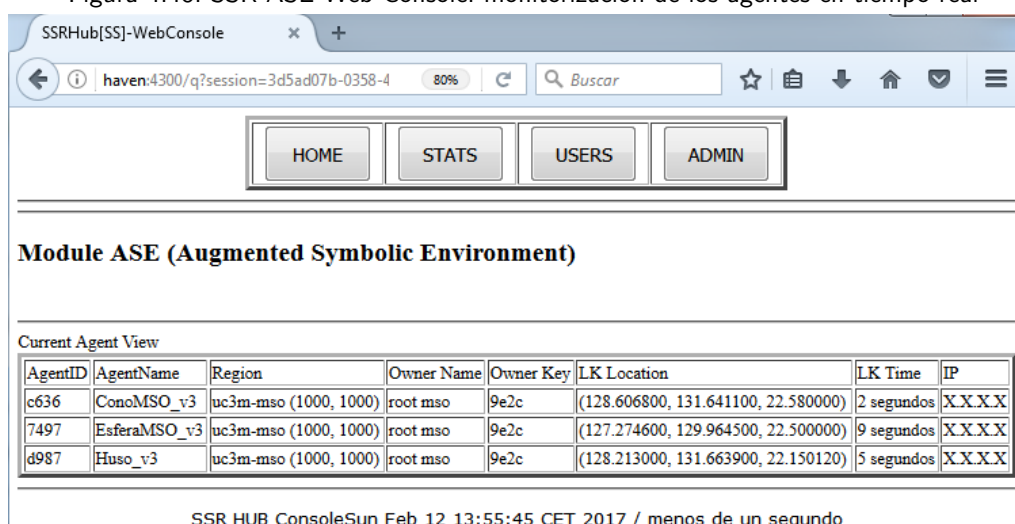
Estas son las lecciones aprendidas y toma de decisiones consecuentes que han ido ocurriendo:

- En el planteamiento original de 2011 se utiliza de una región de acceso público en *Second Life*, a través de una sesión compartida. El resultado, aunque decididamente interesante para los alumnos resultó algo desordenado al trabajar todos los alumnos al mismo tiempo en la misma zona. Adicionalmente, al disponer únicamente de una sola batidora, se produjeron algunos conflictos.
- En el siguiente año (2012) se siguió utilizando la misma aproximación usando *Second Life* aunque se intentó mitigar el desorden del año anterior utilizando un entorno virtual más grande y con un mayor número de batidoras. Aunque el desorden disminuyó, la experiencia siguió resultando complicada de seguir al principio.
- En los siguientes dos años (2013 y 2014) se planteó por primera vez *Open Simulator* en modo standalone, mejorando completamente la experimentación en cuanto a

confusión y compartición de recursos. Ahora cada alumno tenía su propia batidora que podía usar siempre que quisiese. El problema surgió al no tener centralizada la recogida de muestreos ya que cada alumno tenía su propia base de datos. En otras palabras, no se había separado el **DS** y el **SSR**.

- Finalmente, en 2015 y 2016 se ha utilizado con total éxito el modelo *Open Simulator* standalone pero introduciendo en la práctica la última versión del framework **SSR-Hub** y desarrollando de nuevo el módulo MSO según la nueva estructura.

Figura 4.48: SSR-ASE Web Console: monitorización de los agentes en tiempo real



## 4.5.8 Conclusiones

Este experimento / experiencia docente ha resultado muy interesante, y esperamos siga siéndolo, en numerosos aspectos:

**Educación**, hemos introducido a los estudiantes en el uso aplicado de simuladores y laboratorios virtuales.

**Experiencia**, la eficiencia de la experiencia práctica se ha mejorado mediante la implementación de laboratorios virtuales comunes y privados sin problemas.

**Ciencia**, se ha presentado al alumno el uso práctico de la experimentación científica: a partir de una hipótesis inicial, la descripción de un experimento, la implementación

de un laboratorio, la ejecución del experimento, la recolección y análisis de los datos. Y finalmente la elaboración de los resultados de todo el proceso.

**Validación**, de tanto el modelo teórico **DRM** como de la metodología **PIIVE-DM**.

**Eficiencia**, al mostrar la robustez del servicio **SSR-Hub** al soportar la gran carga de trabajo de numerosos **DS**'s lanzando peticiones de manera concurrente.

Año tras año, esta experiencia educativa se enriquece con la participación tanto de profesores como de estudiantes, y se propone ampliar la gama de experimentos así como utilizarlos en otros temas.

Desde el punto de vista de la presente tesis, en este experimento ha permitido probar los siguientes objetivos: *G1, G2a, G2b, G2c, G4, G5, G6, G7x, G8, G9, G10, G11b*

## 4.6 Conclusiones

Las **cuatro** experiencias aportadas se han llevado a cabo en un lapso de tiempo importante (más de siete años) y son el fruto de un proceso iterativo de refinamiento, prueba y error y por supuesto, la colaboración de los directores de la presente Tesis que han sido un gran apoyo en este recorrido.

Volviendo a recapitular cuáles han sido los resultados de todos los experimentos y agrupando los objetivos cualitativos según las hipótesis que nos marcamos en el punto 1.3.2 tendremos la siguiente tabla de scoring:

Goal	SI/2009	SI/2011	OS/2012	OS/2016	TOTAL
G1 - met. ant.: Metáfora del Entorno	1	1	1	1	4
G2a - pr. dis.: Principio de Máxima Verosimilitud	1		1	1	3
G2b - pr. dis.: Validez Ecológica	1		1	1	3
G2c - pr. dis.: Flujo de Compromiso de Presencia	1		1	1	3
G3 - for. abs.: Entorno $E=(S,O,F)$					0
G4 - mfm: Índice de Inmersión					0
G5 - mfm: Índice de Validez Ecológica					0
G6 - tip: Taxonomía de Servicios	1	1	1	1	4
G7a - mts: Modelo DEC2RPM		1	1	1	3
G7b - mts: Modelo DRM		1	1	1	3
G8 - metodología PIIVE-DM		1	1	1	3
G9 - def. serv.: Internos / Externos	1	1	1		3
G10 - plat. serv: Metodología Práctica	1	1	1	1	4
G11a - framework: MetabotLib	1	1			2
G11b - framework: SSR-Hub		1	1	1	3

Figura 4.49: Tabla de Scoring de Objetivos



Si plasmamos la misma información en un gráfico radial nos muestra la siguiente información:

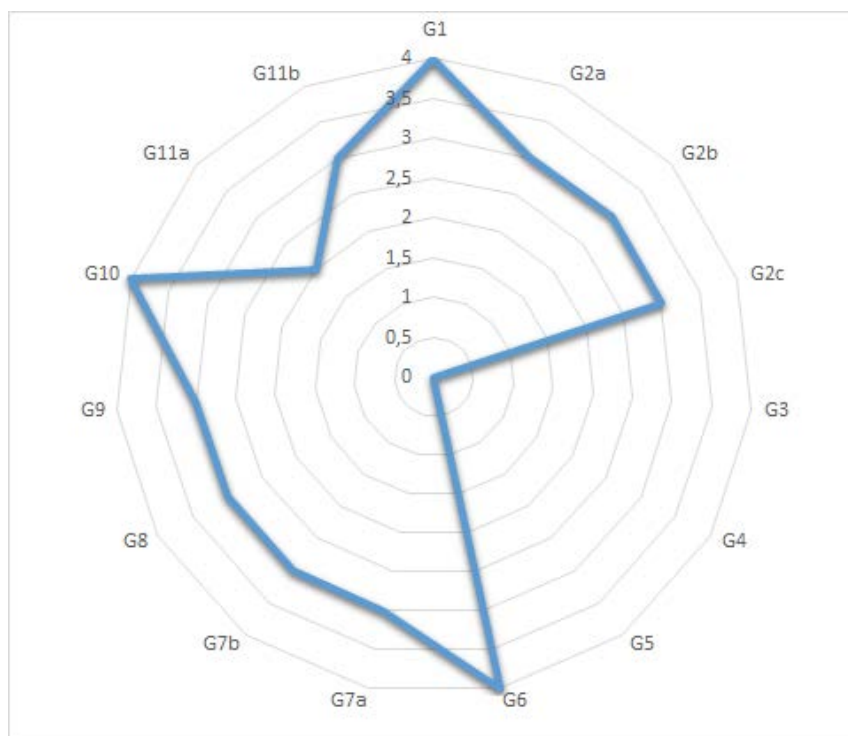


Figura 4.50: Scoring Objetivos



---

# 5

## Conclusiones

*«Hay que tener la mente abierta. Pero no tanto  
como para que se te caiga el cerebro»*

- Richard Feynman

### 5.1 Recapitulación

Llegados a este punto es el momento de hacer recapitulación de lo estudiado en la presente Tesis.

En primer lugar, con el objetivo de centrar y dar un contexto a todo el posterior trabajo desarrollado se ha realizado un estudio práctico sobre los usos e intereses a día de hoy en el campo de la Realidad Virtual. El principal objetivo de dicho estudio es, partiendo de un conjunto de los más importantes sectores económicos existentes, investigar en cuáles de éstos existen casos de uso reales con existan experiencias concretas. Además de revisar uno por uno las distintas publicaciones se elabora un estudio estadístico en función de los casos de uso y los problemas detectados. Con toda esta información se elabora una serie de gráficas y tablas que permite comprender intuitivamente la evolución y los sectores donde la **V.R** tiene uso probado.

Después se realiza un breve resumen de los avances en investigación relativa a *Sistemas Multi-Agente* y en particular de los llamados *Sistemas Multi-Agente Situados* (también llamados **S.M.A.S** de manera abreviada), y de cómo pueden ser estos aplicados de manera eficiente en la construcción de Entornos Virtuales. Aunque en principio pueda parecer que estas dos disciplinas no tienen nada en común, en los siguientes puntos

se propondrá formalmente a los **S.M.A.S** como una herramienta fundamental para el desarrollo de **V.I.E's** y como tal, una solución a los problemas de la **V.R.**

Esta perspectiva se concreta en la última parte del capítulo 2, donde para cada una de las carencias detectadas en los puntos anteriores se repasa el estado del arte al respecto. De esta manera es posible comprender y contextualizar las propuestas mostradas.

En el capítulo 3 se presentan todas las propuestas, en orden lógico de mayor a menor nivel de abstracción. Sin duda el concepto teórico más importante de toda la Tesis es el «Entorno». Podríamos decir que toda la base teórica gira entorno a la definición de este concepto que a pesar de ser un término completamente habitual y cercano al público general (y quizá precisamente por esa razón) tiene numerosas interpretaciones.

El primer problema detectado a la hora de construir entornos **V.R** era la falta de una metáfora y unos principios de diseño adaptados. Basándonos en la «metáfora del escritorio» presentamos la «metáfora del entorno» y los principios de diseño «Validez Ecológica», «Principio de Máxima Verosimilitud» y «Flujo de Compromiso de Presencia» para cubrir esta necesidad primordial.

Posteriormente y dada la relevancia que el término «entorno» tiene se presentaba como tarea imprescindible definirlo de la manera más abstracta y general posible. Para ello se realizaron dos grandes grupos de formalizaciones: el primero fue una formalización abstracta del modelo, y el segundo una taxonomía de servicios.

Durante la formalización abstracta se describió formalmente el entorno, se definieron medidas de similitud entre servicios, se observaron algunos casos particulares de entorno y se propusieron el índice de Inmersión y el índice de validez ecológica, además de dar una definición matemática de lo que es un entorno virtual inmersivo.

Para definir la taxonomía de servicios se realizó una clasificación general de los servicios ecológicos a partir de la observación del mundo real como los principios físicos modernos, la persistencia del entorno o la impenetrabilidad como principio de la interacción. También se identificaron los servicios perceptivos, interactivos y tecnológicos. Y por último una formalización de la notación taxonómica. Una vez afianzadas las bases teóricas ya fue posible construir sobre estas las propuestas arquitectónicas y metodológicas. En primer lugar se definió la arquitectura «**DEC2RPM**», para lo cual se caracterizó el experimento del comportamiento complejo, las interacciones físicas complejas e interacciones inteligentes. También se formaliza la arquitectura basada en los tres canales de interacción y

presentando por primera vez el módulo despliegue representación «**DRM**» como base para la implementación de servicios complejos.

Posteriormente, y ya entrando de lleno en la propuesta de un acercamiento totalmente práctico a la construcción de **V.I.E's** se definió la denominada «Metodología **PIIVE-DM**» que se basa en cuatro pasos: paso 1, definición de requisitos; paso 2, arquitectura de servicios; paso 3, estudio de despliegue; paso 4, diseño de tecnología. También se establecieron herramientas de soporte a la metodología y se sentaron las diferencias entre «proveedores de servicio» y «proveedores de entorno».

Por último, y centrándonos más en la construcción de sistemas multiagente en entornos virtuales inmersivos, se definió la sub-metodología **PIIVE-STC**, y se propuso un framework software denominado «**SSR-Hub**» que se ha utilizado en los experimentos posteriores.

Ya en el capítulo 4 se trataron en exclusiva las experiencias prácticas que han servido para refrendar la efectividad de las propuestas del tema anterior y comprobar que las hipótesis de trabajo establecidas en el capítulo 2 se han visto satisfechas.

En concreto se presentan cuatro experimentos: SL/2009, SL/2011, OS/2012 y OS/2016. En SL/2009 se define el concepto de **Metabot** y se presenta un algoritmo capaz de modelizar y medir la cantidad de atención relativa que está teniendo un conjunto de avatares concretos. Basándose en este algoritmo se experimenta con tres **Metabots** con distintos aspectos y distintas capacidades interactivas para observar qué factores son clave a la hora de medir la inmersión y el compromiso de presencia.

En SL/2011 se propone una estrategia multiagente deliberativa denominada «sprawl1» orientada a la resolución de problemas y comportamientos complejos. Para ello se definió un marco de resolución de problemas, un software capaz de implementar el sistema y un conjunto de experimentos de distinta complejidad. Mediante esta experimentación se pudo observar que la arquitectura era operativa y que los agentes se comportaban de una manera más compleja en el entorno virtual de lo que se podría esperar de un problema deliberativo clásico.

En OS/2012 se propone una estrategia multiagente mixta que se denominó «sprawl2» orientada a la resolución de cualquier tipo de problemas. Para construir un marco genérico de resolución de problemas se definió la herramienta software «seed» y una serie de herramientas necesarias: la función de propagación de la energía, el coeficiente de propagación, la distancia especial, los campos de seeds (seedsets), tags y tagargs. Posteriormente se

plantean dos experimentos, el primero fue una búsqueda y almacenamiento de recursos (antcolony) clásico y el segundo la creación de bandadas dinámicas (flocking). Finalmente se recogen los resultados obtenidos, quedando probada la potencia de la arquitectura y del modelo propuesto.

Por último en OS/2016 nos centramos en realizar y documentar las metodologías propuestas para la construcción de entornos virtuales inmersivos «PIIVE-DM» y «PIIVE-STC». Para ello se eligió utilizar dichas metodologías en la elaboración del laboratorio virtual de una asignatura de Máster que se está realizando actualmente en la Universidad Carlos III. A lo largo de la experiencia se describe el objetivo docente de la práctica así como el objetivo investigador de la misma. Se documentó el resultado de aplicar la metodología y se describen los detalles de la experiencia práctica y del experimento docente. Por último se recapitulan las lecciones aprendidas y se evalúa la idoneidad de la metodología para la experimentación propuesta.

## 5.2 Principales Aportaciones

Podemos resumir las principales aportaciones de la presente Tesis mediante esta representación gráfica:

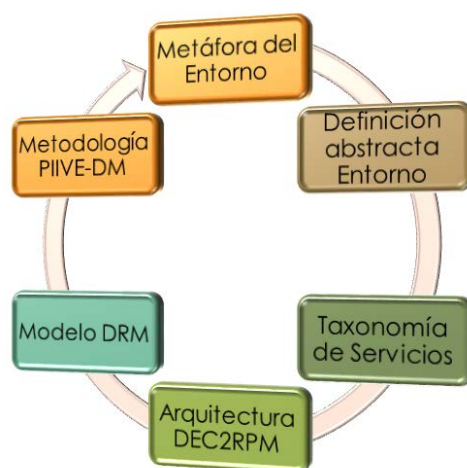


Figura 5.1: Las seis propuestas dentro del ciclo teórico - práctico

Las seis propuestas realizadas van desde la eminentemente conceptual metáfora del entorno hasta el totalmente práctico framework **SSR-Hub**.

Sin embargo cada una de esas seis propuestas está a su vez formada por varios componentes, los más importantes de los cuales se distinguen en la siguiente figura:



Figura 5.2: Las seis propuestas con sus componentes más importantes

## 5.3 Futuros Trabajos

Es mucho lo que queda por hacer en el campo de las metodologías software de la **V.R.** En esta Tesis se ha intentado allanar el terreno, aclarar terminología imprecisa y a veces inexistente y proponer métodos de trabajo, pero apenas hemos dado un primer paso en una senda poco explorada.

Por otro lado es muy improbable que en el futuro no existan cambios tecnológicos de gran calado que puedan modificar completamente al modelo teórico enunciado: la llamada *Ley de Rendimientos Acelerados* o en inglés *Law of accelerated returns* que fue enunciada por Raymond Kurzweil en (Kurzweil, 2004) como una generalización de la *Ley de Moore* (Moore, 1965) a todos los aspectos del progreso tecnológico afirma que «los cambios tecnológicos tan rápidos y profundos que representarán una ruptura en el tejido de la historia humana».

Por estas dos razones: camino por explorar y sujeto a posibles cambios, el modelo teórico propuesto deberá ser revisado y extendido constantemente en caso de ser necesario.

Además de esto, si el objetivo último de esta Tesis es plantear una metodología práctica para para solventar los problemas que se presentan a la hora de construir **V.I.E's**, es necesario probar una y otra vez dicho modelo metodológico con dos objetivos: comprobar su efectividad y aprender de la experiencia.

Por esta razón veo dos posibles ramas de investigación emergentes de esta Tesis: teórica y experimental.

- Teórica:
  - **Modelo formal de «Entorno»**, existe una simplificación voluntaria e implícita en los cálculos: no se ha acotado el volumen del entorno, o dicho de otra manera el volumen de espacio-tiempo que cabe en un simulador. Se han definido todos los entornos como de «tamaño indefinido» pero algunos cálculos preliminares indican que los índices de inmersión y validez ecológica pueden variar.
  - **Taxonomía de Servicios Tecnológicos**, aunque se ha definido su alcance sería posible incluir más elementos en la taxonomía en el momento que realimentemos los resultados prácticos de la aplicación de la metodología.



- **Concepto de Omnicanalidad**, un concepto que no se ha desarrollado en la presente Tesis entronca con la metodología **PIIVE-DM** y podría resumirse como la independización de los servicios a usuarios del canal por el que se transmitan, de menos a más en cuanto a su índice inmersivo: línea telefónica, chat, web, mobile, kiosk, **V.R**, presencia real. Este enfoque podría ampliar el campo de aplicación de la metodología.
  - **Visualización de Datos**. Como ya se indicó en las conclusiones del estado del arte sobre las aplicaciones de la **V.R**, uno de los campos más prometedores en todos los sectores económicos es la visualización de datos, sobre todo asociado al *Big Data*. El hecho de mejorar la manera en la que los seres humanos comprendemos la información compleja es aplicable a la tarea de *toma de decisión*, siempre tan crítica. Existen conceptos como el «*cognitive fit*» que está muy relacionado con la *validez ecológica* y con la calidad de la percepción en general. ¿Podríamos mejorar el entendimiento de los datos arrojados por un proceso de análisis de big data mediante la visualización de datos en **V.R** conectándolo con el modelo teórico aquí presentado?. Esta última pregunta constituye en sí misma una hipótesis de partida, y de ser cierta, podría marcar una nueva y prometedora línea de investigación.
  - **Realidad Aumentada**. Toda la investigación y la experimentación de esta tesis ha sido dirigida a la **V.R**, sin entrar en detalles con la **A.R**. Aunque existen indudables similitudes entre las tecnologías, no son exactamente iguales ni sus casos de uso o problemas son los mismos. Una posible nueva línea de investigación podría ser estudiar qué partes de los modelos teóricos, metodologías y herramientas son directamente transportables a la **A.R**, cuales no, y qué sinergias podemos encontrar.
- Experimental:
    - **Mediciones de Índice Inmersivo e Índice Ecológico**, aunque se ha definido formalmente el algoritmo a alto nivel de cómo deberían ser estas mediciones no se han calculado empíricamente los valores  $\gamma$  ni caracterizado la metodología para su cálculo. Las siguientes experimentaciones que se van a realizar van en este sentido con el objetivo último de generar un proceso científico y regulado capaz de medir cuantitativamente cómo de inmersivo y ecológico es un **V.I.E** dado.

- **Asignatura de Máster, MSO**, la experiencia que se ha mostrado en el capítulo 4.5 se basa en una asignatura que sigue en marcha hoy en día, por lo cual dicho experimento no ha terminado.
- **Metodología *PIIVE-DM***, se requieren más experimentos que permitan probar y aprender de la experiencia en la aplicación a esta metodología. Una posible vía de actuación es buscar la experiencia en la Univesidad Politécnica con los Laboratorios Virtuales en otras entidades docentes universitarias.

---

# A

## Apéndice

### A.1 Proyectos relacionados

A lo largo de los años de investigación en los que se enmarca esta tesis han surgido distintas oportunidades de colaboración con entidades privadas y públicas que se describen a continuación.

#### A.1.1 Babybot

La empresa de capital privado **Optenet** desarrolló un software capaz de imitar el comportamiento verbal de un menor de edad con la capacidad adicional de detectar cuando las conversaciones que estaba desarrollando con interlocutores humanos era interpretable como acoso y un posible caso de pedofilia. Este software estaba orientado al entorno *chat* puro por lo que al intentar ampliar este producto a los entornos virtuales requirieron la capacidad de investir con un avatar a su «*BabyBot*». Para ello utilizaron la **MetabotLib** como software base y por consiguiente parte del análisis teórico que se ha desarrollado en la presente tesis. El proyecto, además de estar apadrinado por la Universidad Politécnica de Madrid, tuvo bastante repercusión en los medios .

#### A.1.2 Laboratorios Virtuales UPM

Dentro del entorno universitario, y en especial en las ciencias experimentales, las necesidades de proveer a los alumnos de un entorno práctico son cada vez más importantes. Como se ha discutido largamente a lo largo de esta tesis, conceptos como la *validez*

*ecológica* son fundamentales para asegurarnos que estas *actividades prácticas* puedan ser transportadas con garantías a los **V.I.E's**. Como se puede observar en el experimento del capítulo 4.5 dentro de las ciencias informáticas es más o menos sencillo, sin embargo afrontar el reto de implementar *laboratorios virtuales* de disciplinas no computacionales requiere un alto nivel de dedicación.

La *Universidad Politécnica de Madrid «UPM»* y en particular el *Gabinete de Tele-Educación «GATE»* comenzó en 2012 a interesarse por la posibilidad de implementar una plataforma virtual que fuese potencialmente utilizable por cualquier facultad o escuela independientemente de su naturaleza y tras ponerse en contacto con el equipo que habíamos implementado el Concurso de Programación de Sistemas Inteligentes Virtuales - UPM junto con otras experiencias incipientes similares se inició mi participación personal como asesor desde esa fecha hasta la actualidad.

El resultado de estos cinco años de trabajo y de un equipo excepcional de diseñadores, programadores y técnicos ha dado como resultado laboratorios operativos en estudios tan dispares como: química, biotecnología agroforestal, patología forestal, electrónica, materiales, biología, topografía, etc.

La experiencia práctica en este proyecto ha sido uno de los campos de pruebas más exigentes y útiles para la metodología **PIIVE-DM** que se ha presentado en el capítulo 3.7 de este tesis.

En este momento los *UPM 3DLabs* <https://3dlabs.upm.es> están dando servicio a seis regiones en producción con acceso libre y cinco regiones en construcción con acceso limitado. El acceso libre significa que cualquier alumno de la *UPM* que disponga de cuenta de correo *UPM* puede solicitar un avatar y experimentar como visitante cualquiera de esos laboratorios virtuales. Obviamente únicamente los alumnos registrados en las asignaturas correspondientes tienen permiso para realizar las prácticas, siempre con el soporte virtual de sus tutores de la asignatura.

## A.2 Publicaciones relacionadas

Durante el desarrollo de la presente tesis se han publicado distintas aportaciones a conferencias y publicaciones en revistas.

### A.2.1 Publicaciones en revistas internacionales (JCR)

- **2011** Arroyo, A., Serradilla, F., and Calvo, O. (2011). Adaptive fuzzy knowledge-based systems for control metabots' mobility on virtual environments. *Expert Systems* 28.4 (2011) 339–352

### A.2.2 Publicaciones en conferencias

- **2017** O. Calvo, Jose M. Molina, Miguel A. Patricio, A. Berlanga (2017). A Propose Architecture for Situated Multi-agent Systems and Virtual Simulated Environments applied to Educational Immersive Experiences. *International Work-Conference on the Interplay Between Natural and Artificial Computation*
- **2009** Arroyo, A., Serradilla, F., Calvo, O. Springer (2009). Multimodal agents in second life and the new agents of virtual 3d environments. pages 506–516. *International Work-Conference on the Interplay Between Natural and Artificial Computation*
- **2010** Ángel Arroyo, Francisco Serradilla, Óscar Calvo. Modeling spatial-temporal context information in virtual worlds. *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer Berlin Heidelberg

### A.3 Acceso a la última versión del documento

Es posible solicitar una copia digital de la presente tesis en la siguiente dirección:

<http://raxolabs.duckdns.org/tesis/request.html>



Figura A.1: TesisQR

También es posible conseguir más información sobre otros trabajos en la página web del autor: <http://raxolabs.duckdns.org/oscar/index.html>

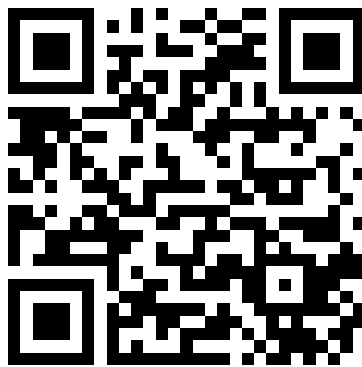


Figura A.2: Página Web de Oscar Calvo

---

## Bibliografía

Alewine, N. J., Rick, A. H. I., Hood, R. A., and Seaman, J. W. (2016). System and method for detecting inappropriate content in virtual worlds. US Patent App. 15/212,393.

Arkin, R. C. (1989). Motor schema—based mobile robot navigation. *The International journal of robotics research*, 8(4):92–112.

Armstrong, C. M., Reger, G. M., Edwards, J., Rizzo, A. A., Courtney, C. G., and Parsons, T. D. (2013). Validity of the virtual reality stroop task (vrst) in active duty military. 35:113–123.

Arroyo, A., Serradilla, F., and Calvo, O. (2011). Adaptive fuzzy knowledge-based systems for control metabots' mobility on virtual environments. 28:339–352.

Arroyo, A., Serradilla, F., Calvo, O., and Springer (2009). Multimodal agents in second life and the new agents of virtual 3d environments. pages 506–516. International Work-Conference on the Interplay Between Natural and Artificial Computation.

Arthur, R. (2016). Future of retail: Artificial intelligence and virtual reality have big roles to play. *Forbes*.

Baldominos Gómez, A., Sáez Achaerandio, Y., and García del Pozo, M. C. (2015). An approach to physical rehabilitation using state-of-the-art virtual reality and motion tracking technologies.

Bandini, S., Manzoni, S., Pavesi, G., and Simone, C. (2001). Location of extra-urban shopping centres: A multi-agent based approach. *Proceedings of INPUT*.

Bandini, S., Manzoni, S., and Simone, C. (2002). Dealing with space in multi-agent systems: a model for situated mas. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 3*, pages 1183–1190. ACM.

Beckmann, P. (2015). *A history of pi*. St. Martin's Griffin.

Bellifemine, F., Poggi, A., and Rimassa, G. (2001). Developing multi-agent systems with a fipa-compliant agent framework. *Software-Practice and experience*, 31(2):103–128.

Bentham, J. and Dumont, E. (1840). *Theory of legislation*, volume 1. Weeks, Jordan,.

- Berg, L. P. and Vance, J. M. (2016). Industry use of virtual reality in product design and manufacturing: a survey. pages 1–17.
- Berman, B. and Thelen, S. (2004). A guide to developing and managing a well-integrated multi-channel retail strategy. *International Journal of Retail & Distribution Management*, 32(3):147–156.
- Bombín-González, I., Cifuentes-Rodríguez, A., Climent-Martínez, G., Luna-Lario, P., Cardas-Ibáñez, J., Tirapu-Ustárroz, J., and Díaz-Orueta, U. (2014). Validez ecológica y entornos multitarea en la evaluación de las funciones ejecutivas. *Rev Neurol*, 59(2):77–87.
- Bourbaki, N. (1998). General topology. chapters 1–4, elements of mathematics. *SpringerVerlag, Berlin*, 6:1–3.
- Bricken, M. and Byrne, C. M. (1993). Summer students in virtual reality. pages 199–218.
- Brooks, F. P. (1999). What’s real about virtual reality? *IEEE Computer graphics and applications*, 19(6):16–27.
- Burdea, G. C. (1999). Invited review: the synergy between virtual reality and robotics. 15:400–410.
- Carr, D. (2006). *Computer games: Text, narrative and play*. Polity.
- Choi, H.-s. and Kim, S.-h. (2017). A content service deployment plan for metaverse museum exhibitions—centering on the combination of beacons and hmds. *International Journal of Information Management*, 37(1):1519–1527.
- Cooper, A. et al. (2004). *The inmates are running the asylum:[Why high-tech products drive us crazy and how to restore the sanity]*. Sams Indianapolis, IN, USA:.
- Corchado, J. M. and Molina, J. M. (2002). *Introducción a la teoría de agentes y sistemas multiagente*. Catedral.
- Couclelis, H. (1997). From cellular automata to urban models: new principles for model development and implementation. *Environment and planning B: Planning and design*, 24(2):165–174.
- Cowan, B. and Kapralos, B. (2017). An overview of serious game engines and frameworks. In *Recent Advances in Technologies for Inclusive Well-Being*, pages 15–38. Springer.
- Cranefield, S. and Li, G. (2010). Monitoring social expectations in second life. In *Coordination, Organizations, Institutions and Norms in Agent Systems V*, pages 133–146. Springer.



- Deneubourg, J. L., Aron, S., Goss, S., Pasteels, J., and Duerinck, G. (1986). Random behaviour, amplification processes and number of participants: how they contribute to the foraging properties of ants. *Physica D: nonlinear phenomena*, 22(1-3):176–186.
- Dholakia, U. M., Kahn, B. E., Reeves, R., Rindfleisch, A., Stewart, D., and Taylor, E. (2010). Consumer behavior in a multichannel, multimedia retailing environment. *Journal of Interactive Marketing*, 24(2):86–95.
- Dijkstra, E. W. (1972). The humble programmer. *Commun. ACM*, 15(10):859–866.
- Dorigo, M., Birattari, M., and Stutzle, T. (2006). Ant colony optimization. *IEEE computational intelligence magazine*, 1(4):28–39.
- Dorigo, M. and Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation*, 1(1):53–66.
- Dorigo, M., Maniezzo, V., and Coloni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1):29–41.
- Drogoul, A. and Ferber, J. (1992). Multi-agent simulation as a tool for modeling societies: Application to social differentiation in ant colonies. In *European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, pages 2–23. Springer.
- Ebert, C. (1997). The road to maturity: navigating between craft and science. *IEEE Software*, 14(6):77–82.
- Erath, A., Maheshwari, T., Joos, M., Kupferschmid, J., and van Eggermond, M. (2016). Visualizing transport futures: the potential of integrating procedural 3d modelling and traffic micro-simulation in virtual reality applications. *Arbeitsberichte Verkehrs-und Raumplanung*, 1185.
- Erleben, K. (2004). Stable, robust, and versatile multibody dynamics animation. *Unpublished Ph. D. Thesis, University of Copenhagen, Copenhagen*.
- Fenn, J. and LeHong, H. (2011). Hype cycle for emerging technologies, 2011. *Gartner, July*.
- Ferber, J. (1999). *Multi-agent systems: an introduction to distributed artificial intelligence*, volume 1. Addison-Wesley Reading.
- Ferber, J. and Müller, J.-P. (1996). Influences and reaction: a model of situated multiagent systems. In *Proceedings of Second International Conference on Multi-Agent Systems (ICMAS-96)*, pages 72–79.

- Feyerabend, P. K. (1986). Tratado contra el método.
- Feynman, R. P. (1982). Simulating physics with computers. *International journal of theoretical physics*, 21(6):467–488.
- FIPA (2002). Fipa. foundation for intelligent physical agents, fipa abstract architecture specification. FIPA.
- Flanagan, J. (2014). Virtual reality and retail: Fashion's ecommerce trends. *huffingtonpost.com*.
- Foster, R. N. (1986). Working the s-curve: assessing technological threats. *Research Management*, 29(4):17–20.
- Friedman, D., Steed, A., and Slater, M. (2007). Spatial social behavior in second life. In *Intelligent virtual agents*, pages 252–263. Springer.
- Fuhrmann, A. and Schmalstieg, D. (1999). Concept and implementation of a collaborative workspace for augmented reality.
- Gibson, W. (1984). *Neuromancer*. Ace.
- Gros, B. (2016). Game dimensions and pedagogical dimension in serious games. *Handbook of Research on Serious Games for Educational Applications*, page 402.
- Guidi, G., Micoli, L. L., Casagrande, C., Ghezzi, L., and IEEE (2010). Virtual reality for retail. pages 285–288. Virtual Systems and Multimedia (VSMM), 2010 16th International Conference on.
- Guttentag, D. A. (2010). Virtual reality: Applications and implications for tourism. 31:637–651.
- Helander, M. G. (2014). *Handbook of human-computer interaction*. Elsevier.
- Helleboogh, A., Vizzari, G., Uhrmacher, A., and Michel, F. (2007). Modeling dynamic environments in multi-agent simulation. *Autonomous Agents and Multi-Agent Systems*, 14(1):87–116.
- Huang, Y. C., Backman, K. F., Backman, S. J., and Chang, L. L. (2016). Exploring the implications of virtual reality technology in tourism marketing: An integrated research framework. *International Journal of Tourism Research*, 18(2):116–128.
- Hwang, J., Yoon, S.-Y., and Bendle, L. J. (2012). Desired privacy and the impact of crowding on customer emotions and approach-avoidance responses: Waiting in a virtual reality restaurant. 24:224–250.

- Julier, S., Lindeman, R., and Sandor, C. (2015). Guest editor's introduction to the special section on the IEEE international symposium on mixed and augmented reality 2014. *IEEE Transactions on Visualization and Computer Graphics*, 21(12):1321–1322.
- Kade, D., Akcsit, K., Ürey, H., and Özcan, O. (2015). Head-mounted mixed reality projection display for games production and entertainment. 19:509–521.
- Katie Drummond, Ellis Hamburger, T. H. T. I. U. T. R. L. D. L. C. M. C. N. A. R. M. S. M. S. S. T. and Zelenko., M. (2014). The rise and fall and rise of virtual reality. *The Verge* <http://www.theverge.com/a/virtual-reality>.
- Kaufmann, H., Schmalstieg, D., and Wagner, M. (2000). Construct3d: a virtual reality application for mathematics and geometry education. 5:263–276.
- Kim, G. J. et al. (2005). A swot analysis of the field of virtual reality rehabilitation and therapy. 14:119–146.
- Krummel, T. M. (1998). Surgical simulation and virtual reality: the coming revolution. *Annals of surgery*, 228(5):635.
- Kurzweil, R. (2004). The law of accelerating returns. In *Alan Turing: Life and Legacy of a Great Thinker*, pages 381–416. Springer.
- Lele, A. (2013). Virtual reality and its military utility. 4:17–26.
- Li, X., Lv, Z., Wang, W., Wu, C., and Hu, J. (2015). Virtual reality gis and cloud service based traffic analysis platform. In *Geoinformatics, 2015 23rd International Conference on*, pages 1–6. IEEE.
- Linden, A. and Fenn, J. (2003). Understanding gartner's hype cycles. *Strategic Analysis Report N° R-20-1971. Gartner, Inc.*
- Lui, T.-W., Piccoli, G., and Ives, B. (2007). Marketing strategies in virtual worlds. 38:77–80.
- Maes, P. and Maes, P. (1994). Modeling adaptive autonomous agents. In *Artificial Life*. MIT Press.
- Mamei, M. and Zambonelli, F. (2004). Motion coordination in the quake 3 arena environment: a field-based approach. In *International Workshop on Environments for Multi-Agent Systems*, pages 264–278. Springer.
- McCarthy, J. and Hayes, P. J. (1969). Some philosophical problems from the standpoint of artificial intelligence. *Readings in artificial intelligence*, pages 431–450.

- McCarthy, J., Minsky, M., and Rochester, N. (1955). A proposal for the dartmouth summer research project on artificial intelligence. *DARTMOUTH SUMMER RESEARCH PROJECT*.
- McMahan, A. (2003). Immersion, engagement and presence. *The video game theory reader*, 67:86.
- Michael, D. R. and Chen, S. L. (2005). *Serious games: Games that educate, train, and inform*. Muska & Lipman/Premier-Trade.
- Milgram, P., Takemura, H., Utsumi, A., and Kishino, F. (1995). Augmented reality: A class of displays on the reality-virtuality continuum. In *Photonics for industrial applications*, pages 282–292. International Society for Optics and Photonics.
- Mohan, S. and Laird, J. E. (2009). Learning to play mario. *Tech. Rep. CCA-TR-2009-03*.
- Moore, G. (1965). Moore's law. *Electronics Magazine*, 38(8):114.
- Mujber, T. S., Szecsi, T., and Hashmi, M. S. (2004). Virtual reality applications in manufacturing process simulation. 155:1834–1838.
- Nevelsteen, K. J. (2015). *A Survey of Characteristic Engine Features for Technology-Sustained Pervasive Games*. Springer.
- Newell, A. and Shaw, J. (1959). A variety of intelligent learning in a general problem solver. *RAND Report P-1742, dated July*, 6.
- Newell, A., Shaw, J. C., and Simon, H. A. (1958). Elements of a theory of human problem solving. *Psychological review*, 65(3):151.
- Newell, A., Simon, H. A., et al. (1972). *Human problem solving*, volume 104. Prentice-Hall Englewood Cliffs, NJ.
- Ángel Arroyo, Serradilla, F., Óscar Calvo, and Springer (2010). Modeling spatial-temporal context information in virtual worlds. pages 437–447. International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems.
- Nguyen, M.-T., Nguyen, H.-K., Vo-Lam, K.-D., Nguyen, X.-G., and Tran, M.-T. (2016). Applying virtual reality in city planning. In *International Conference on Virtual, Augmented and Mixed Reality*, pages 724–735. Springer.
- North Atlantic Treaty Organisation, R. and Organisation, T. (2003). Virtual reality: State of military research and applications in member countries.
- Odell, J., Parunak, H. V. D., and Fleischer, M. (2003). Modeling agents and their environment: the communication environment. *Journal of Object Technology*, 2(1):39–52.

- Omicini, A. (2001). Soda: Societies and infrastructures in the analysis and design of agent-based systems. In *Agent-oriented software engineering*, pages 185–193. Springer.
- Otto, J. R. and Chung, Q. (2000). A framework for cyber-enhanced retailing: Integrating e-commerce retailing with brick-and-mortar retailing. *Electronic Markets*, 10(3):185–191.
- Parsons, T. D. (2015). Ecological validity in virtual reality-based neuropsychological assessment. In *Encyclopedia of Information Science and Technology, Third Edition*, pages 1006–1015. IGI global.
- Parsons, T. D., Rizzo, A. A., Rogers, S., and York, P. (2009). Virtual reality in paediatric rehabilitation: A review. 12:224–238.
- Parunak, H. V. D. (1997). "go to the ant": Engineering principles from natural multi-agent systems. *Annals of Operations Research*, 75:69–101.
- Pimentel, K. and Teixeira, K. (1993). Virtual reality through the new looking glass.
- Platon, E., Mamei, M., Sabouret, N., Honiden, S., and Parunak, H. V. D. (2007). Mechanisms for environments in multi-agent systems: Survey and opportunities. *Autonomous Agents and Multi-Agent Systems*, 14(1):31–47.
- Popescu, V. G., Burdea, G. C., Bouzit, M., and Hentz, V. R. (2000). A virtual-reality-based telerehabilitation system with force feedback. *IEEE transactions on Information Technology in Biomedicine*, 4(1):45–51.
- Portillo, M. T. E. and Mendoza, J. A. S. P. (2015). P. ch. mahalanobis y las aplicaciones de su distancia estadística. *CULCyT*, (27).
- Pruitt, J. and Grudin, J. (2003). Personas: practice and theory. In *Proceedings of the 2003 conference on Designing for user experiences*, pages 1–15. ACM.
- Reiter, R. (1978). *On closed world data bases*. Plenum Publ. Co.
- Reiter, R. (1991). *The frame problem in the situation calculus: A simple solution and a completeness result for goal regression*. Academic Press.
- Renz, J. and Ge, X. (2017). Physics simulation games. *Handbook of Digital Games and Entertainment Technologies*, pages 77–95.
- Ricci, A., Omicini, A., and Denti, E. (2002). Activity theory as a framework for mas coordination. In *International Workshop on Engineering Societies in the Agents World*, pages 96–110. Springer.

- Ritchings, T., Anastassakis, G., and Panayiotopoulos, T. (2001). Multi-agent systems as intelligent virtual environments. In *Annual Conference on Artificial Intelligence*, pages 381–395. Springer.
- Rizzo, A. A., Bowerly, T., Buckwalter, J. G., Klimchuk, D., Mitura, R., and Parsons, T. D. (2006). A virtual reality scenario for all seasons: the virtual classroom. 11:35–44.
- Rizzo, A. A. and Buckwalter, J. G. (1997). Virtual reality and cognitive assessment. 44:123.
- Rizzo, A. A., Buckwalter, J. G., Bowerly, T., Van Der Zaag, C., Humphrey, L., Neumann, U., Chua, C., Kyriakakis, C., Van Rooyen, A., and Sisemore, D. (2000). The virtual classroom: a virtual reality environment for the assessment and rehabilitation of attention deficits. *CyberPsychology & Behavior*, 3(3):483–499.
- Rizzo, A. A., Schultheis, M., Kerns, K. A., and Mateer, C. (2004). Analysis of assets for virtual reality applications in neuropsychology. 14:207–239.
- Rothbaum, B. O., Price, M., Jovanovic, T., Norrholm, S. D., Gerardi, M., Dunlop, B., Davis, M., Bradley, B., Duncan, E. J., Rizzo, A., et al. (2014). A randomized, double-blind evaluation of d-cycloserine or alprazolam combined with virtual reality exposure therapy for posttraumatic stress disorder in iraq and afghanistan war veterans. *American Journal of Psychiatry*, 171(6):640–648.
- Ruddle, R. A. (2004). The effect of environment characteristics and user interaction on levels of virtual environment sickness. In *Virtual Reality, 2004. Proceedings. IEEE*, pages 141–285. IEEE.
- Runde, V. (2006). *A Taste of Topology*. Universitext. Springer New York.
- Russell, S. J., Norvig, P., Canny, J. F., Malik, J. M., and Edwards, D. D. (2003). *Artificial intelligence: a modern approach*, volume 2. Prentice hall Upper Saddle River.
- Satava, R. M. and Jones, S. B. (1998). Current and future applications of virtual reality for medicine. 86:484–489.
- Schmalstieg, D., Fuhrmann, A., Szalavári, Z., and Gervautz, M. (1996). Studierstube—an environment for collaboration in augmented reality. In *CVE'96 Workshop Proceedings*, volume 19.
- Simon, H. (1956). The logic theory machine—a complex information processing system. *IEEE Transactions on Information Theory*, 2(3):61–79.

- Slater, M., Lotto, B., Arnold, M. M., and Sanchez-Vives, M. V. (2009). How we experience immersive virtual environments: the concept of presence and its measurement. *Anuario de psicología/The UB Journal of psychology*, 40(2):193–210.
- Steels, L. (1990). Cooperation between distributed agents through self-organisation. In *Intelligent Robots and Systems' 90.'Towards a New Frontier of Applications', Proceedings. IROS'90. IEEE International Workshop on*, pages 8–14. IEEE.
- Stephenson, N. (1992). *Snow Crash*. Bantam Books (USA).
- Stork, A., Sevilmis, N., Weber, D., Gorecky, D., Stahl, C., Loskyll, M., Michel, F., and IEEE (2012). Enabling virtual assembly training in and beyond the automotive industry. pages 347–352. *Virtual Systems and Multimedia (VSMM)*, 2012 18th International Conference on.
- Szalavári, Z., Schmalstieg, D., Fuhrmann, A., and Gervautz, M. (1998). .studierstube.: An environment for collaboration in augmented reality. 3:37–48.
- Tanriverdi, V. and Jacob, R. J. (2001). Vrid: a design model and methodology for developing virtual reality interfaces. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 175–182. ACM.
- Tarnanas, I., Schlee, W., Tsolaki, M., Müri, R., Mosimann, U., and Nef, T. (2013). Ecological validity of virtual reality daily living activities screening for early dementia: longitudinal study. *JMIR Serious Games*, 1(1):e1.
- Tarr, M. J. and Warren, W. H. (2002). Virtual reality in behavioral neuroscience and beyond. 5:1089–1092.
- Tötösy de Zepetnek, S. and Sywenky, I. (2010). The systemic and empirical approach to literature and culture as theory and application.
- Tussyadiah, I., Wang, D., and Jia, C. H. (2016). Exploring the persuasive power of virtual reality imagery for destination marketing.
- Valle, F. (1985). El problema de la validez ecológica. *Estudios de Psicología*, 6(23-24):135–151.
- Vinge, V. (1981). *True Names*. Penguin Worlds. Penguin Books Limited.
- Viroli, M., Ricci, A., Zambonelli, F., Holvoet, T., and Shelfthout, K. (2005). Engineering the environment of multiagent systems. *Autonomous Agents and Multi-Agent Systems*.
- Wang, R., Wang, X., and Kim, M. J. (2011). Motivated learning agent model for distributed collaborative systems. *Expert Systems with Applications*, 38(2):1079–1088.

- Weyns, D. (2010). Architecture-based design of multi-agent systems.
- Weyns, D., Helleboogh, A., Holvoet, T., and Schumacher, M. (2009). The agent environment in multi-agent systems: A middleware perspective. *Multiagent and Grid Systems*, 5(1):93–108.
- Weyns, D. and Holvoet, T. (2004). A formal model for situated multi-agent systems. *Fundamenta Informaticae*, 63(2-3):125–158.
- Weyns, D. and Michel, F. (2015a). *Agent Environments for Multi-agent Systems – A Research Roadmap*, pages 3–21. Springer International Publishing, Cham.
- Weyns, D. and Michel, F. (2015b). *Agent Environments for Multi-Agent Systems IV: 4th International Workshop, E4MAS 2014 - 10 Years Later, Paris, France, May 6, 2014, Revised Selected and Invited Papers*. Lecture Notes in Computer Science. Springer International Publishing.
- Weyns, D., Omicini, A., and Odell, J. (2007a). Environment as a first class abstraction in multiagent systems. *Autonomous agents and multi-agent systems*, 14(1):5–30.
- Weyns, D., Van Dyke Parunak, H., and Michel, F. (2005). *Environments for Multi-Agent Systems: First International Workshop, E4MAS, 2004, New York, NY, July 19, 2004, Revised Selected Papers*. Lecture Notes in Computer Science. Springer Berlin Heidelberg.
- Weyns, D., Van Dyke Parunak, H., and Michel, F. (2006). *Environments for Multi-Agent Systems II: Second International Workshop, E4MAS 2005, Utrecht, The Netherlands, July 25, 2005, Selected Revised and Invited Papers*. Lecture Notes in Computer Science. Springer Berlin Heidelberg.
- Weyns, D., Van Dyke Parunak, H., and Michel, F. (2007b). *Environments for Multi-Agent Systems III: Third International Workshop, E4MAS 2006, Hakodate, Japan, May 8, 2006, Selected Revised and Invited Papers*. Lecture Notes in Computer Science. Springer Berlin Heidelberg.
- Williams, G. (1984). Apple macintosh computer. *Byte*, 9(2):30–31.
- Wooldridge, M. (2002). *An introduction to multiagent systems*. John Wiley & Sons.
- Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The knowledge engineering review*, 10(02):115–152.
- Wyk, E. V., Villiers, R. D., and ACM (2009). Virtual reality training applications for the mining industry. pages 53–63. Conference Proceedings of the 6th international conference on computer graphics, virtual reality, visualisation and interaction in Africa.



- Xu, F., Buhalis, D., and Weber, J. (2017). Serious games and the gamification of tourism. *Tourism Management*, 60:244–256.
- Yu, Y., El Kamel, A., and Gong, G. (2013). Modeling intelligent vehicle agent in virtual reality traffic simulation system. In *Systems and Computer Science (ICSCS), 2013 2nd International Conference on*, pages 274–279. IEEE.
- Yu, Y., El Kamel, A., Gong, G., and Li, F. (2014). Multi-agent based modeling and simulation of microscopic traffic in virtual reality system. *Simulation Modelling Practice and Theory*, 45:62–79.
- Zhao, J. and Moere, A. V. (2008). Embodiment in data sculpture: a model of the physical visualization of information. In *Proceedings of the 3rd international conference on Digital Interactive Media in Entertainment and Arts*, pages 343–350. ACM.
- Zyda, M. (2005). From visual simulation to virtual reality to games. 38:25–32.